An Introduction to MOSES

Spring 2013 Rev C.3

Contents

1 Introduction

The purpose of this document is to provide a gentle introduction to the MOSES software package. It is intended as a complement to the MOSES manual, example and test files, and the web page. The approach taken here is to:

- discuss an example or a test
- ask the user to change a few entries, and
- discuss the changes in the results

All of the files needed to complete the exercises are contained in the standard MOSES distribution. It is a good habit to copy these files to a working directory so that you can always revert to the original. It is up to you to decide to copy the files or work within the ultra directory. There is a word of caution to be said here:

> When you receive a MOSES update, or you decide to download a new MOSES version, the installation wizard will write over any files in the ultra directory. **If you choose to work in the ultra directory, you run the risk of losing your work.**

For each exercise I tried to have a set of topics, reference files, other commands to use, and a purpose before the discussion. The discussion covers by example the topics listed. The reference files were used to present the topics. In most cases, the reference files are part of the standard MOSES distribution.

Text Editor

You will need to view and change the text of many files. You are welcome to use any editor of your choice. Popular text editors are WordPad, VI, Crimson Editor, Medit, Ultra–edit, and Emacs. It is assumed that you know how to use the editor you choose and no attempt will be made to try to teach how to use an editor.

I do make references to line numbers. It will be easier to follow if your text editor shows line numbers.

1.1 Installing MOSES - Windows

The following are the standard set of instructions for installing MOSES. If you would like to watch a video of this, please go to the following site:

http://bentley.ultramarine.com/pub/download.htm

The section **Install MOSES:** has an entry for Windows machine. Please click were indicated.

On a Standalone Machine

- CD-ROM Versions
  - Insert the CD. It should start the install procedure. If not, go to the CD–ROM drive (often D:) and double–click on setup.exe.
- Download Versions
  - Locate and run the downloaded file. It should be called something similar to moses_download_win32.exe.
- Both Versions
  - Press Next until you get to the "Choose Components" screen.
  - If you want to keep your old MOSES install, select "Backup old files." This will create a directory ultra_p with your old MOSES install and settings.
  - Associating Files registers MOSES's CIF and DAT extensions with Windows so you can double click on them. This is recommended for most machines, but is not necessary for installs on a file server.
  - Do not install the Sentinel Hardkey Driver unless you are using a Sentinel Hardkey and this is the only program that is using it. Otherwise, this can conflict with previously installed drivers.
  - Allow the installer to complete.
  - You should now be able to double click on any .cif file on the machine and have MOSES run.

On a Networked Machine (instructions for IT staff and advanced users)

On the File Server:

- Install the software by following the steps above.
- Change the permissions as follows:
  - \ultra −read (all files and subdirectories)
  - \ultra\data\progm −read−write
  - \ultra\data\site −read−write
- Share the \ultra directory or a parent directory of \ultra

On the users' machines:

- Mount the \ultra directory as a drive; we will use U: in this example.
- There should be a moses.exe under U:\ .
- Double click (run) moses.exe. This will bring up a MOSES window.
- MOSES will ask for a file name. Use the name "cow." (cow without the quotes)
- After a few seconds, the main menu will appear at the top of the screen.
- Use the pull down CUSTOMIZE menu.
- Select Register with OS.
- Close MOSES by typing &FINI in the command prompt.
- You should now be able to double click on any .cif file on the machine and have MOSES run.

2 Hydrostatic and Lontudinal Strength

2.1 Getting Started Exercise

Topic:

- Introduction to basic MOSES commands.
- Demonstrate how to restart an analysis and make modifications.

Commands to use: MOSES b_run

Setup Analysis

Run ultra/hdesk/started/b_run/b_run.cif and b_run.dat

Discussion: Running MOSES

In this exercise, the student will become familiar with the file structure and run a simple analysis. At this point, we are not interested in understanding every command, just the concept of menus.

For a list and discussion of the commands found in most examples please see:

http://bentley.ultramarine.com/hdesk/runs/c_htm/common.htm

A discussion and screen shots of the process are also presented at the following link:

http://bentley.ultramarine.com/hdesk/b_run/b_run.htm

If using Windows, you should double click on the file b_run.cif. The file b_run.cif should have an icon that looks like a parabolic shape inside of a half rectangle. Once you double click the CIF file, the MOSES window should appear and the analysis commands scroll by. This should take a few minutes, at most. When the MOSES window disappears, there should now be a b_run.ans and a b_run.dba directory.

The DBA directory is where the database is located. All of the files in this directory are for computers, i.e. not for human eyes. The files in the ANS directory contain the answers (ANS is short for answers).

At the conclusion of the MOSES analysis, we will normally look at the log and the out file, which can both be found in the ANS directory. The log file is a log of the commands used to perform the analysis. The out file is the results of the calculations.

In most exercises, you will be asked to delete the ANS and DBA directory to recreate the answers in the results file or to view the results after data has been changed. At other times, you will be asked to "restart the run," this means you should access the existing database. You "restart the run" by double clicking on the CIF file. You

should not delete the ANS directory unless you are asked to do so.

Discussion: The Analysis

In this analysis, a vessel is set at a draft and trim, the weight necessary to be at equilibrium is computed, then three sets of hydrostatic calculations are reported.

Three things are done in this analysis:

> The curves of form are computed with the section beginning with the command *cform*
>
> Stability righting arm and heeling arm curves are computed with the section beginning with the command *rarm*
>
> Longitudinal strength is computed with the section beginning with the command *equi_h*

Notice that the comments in the CIF file tell us we are entering the hydrostatics menu. Once inside the hydrostatics menu, we can perform hydrostatic calculations. When MOSES is working, if you look at the upper blue bar, you will notice that the words change from "Main Menu" to "Hydrostatic Menu" and "Disposition Menu."

Notice that after each *report* command, there is an *end* command. After each calculation, MOSES is in the disposition menu. When we ask MOSES to *report*, we are asking MOSES to dispose of the results. Once we have finished reporting, we *end* that section of the calculations.

After each of the calculations, a plot is generated with the command *plot*. The numbers after the command *plot* tell MOSES what variables to use as the ordinate and the abscissa. We can edit this file and add the command *vlist* before or after the command *plot* to get a map of the number to variable used.

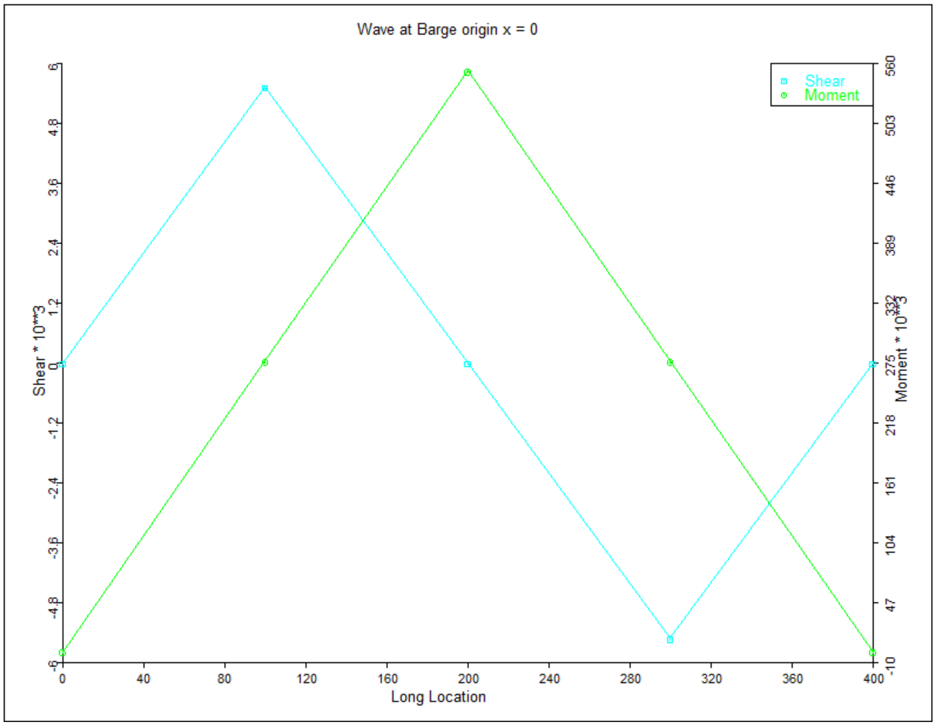


Figure 1: Hog case for basic run

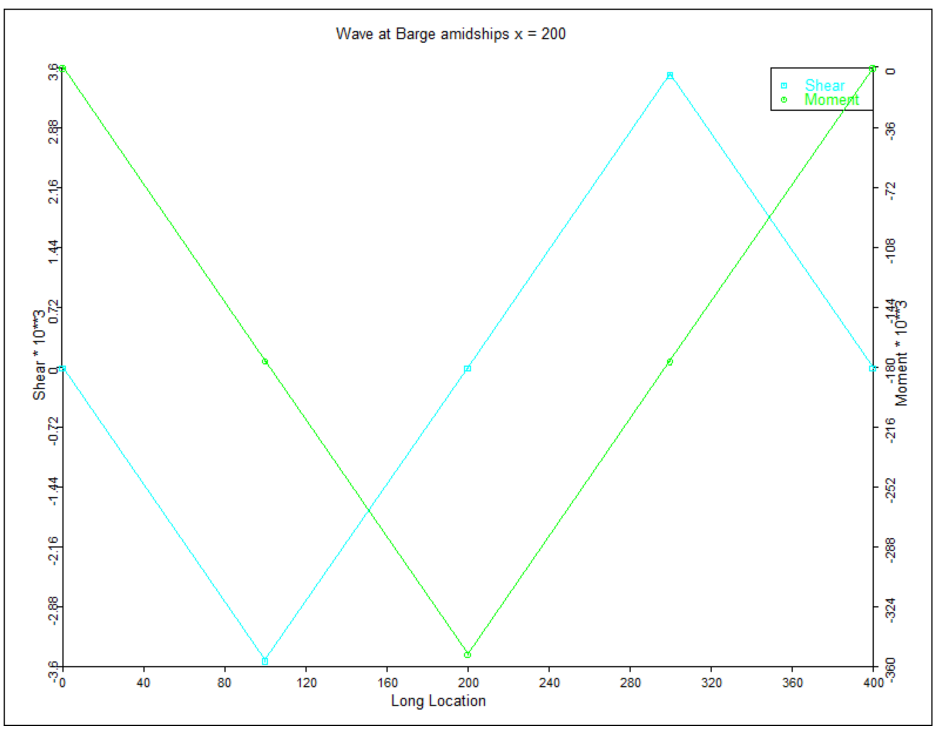


Figure 2: Sag case for basic run

Exercise A

Answer the following questions. The answers are found in the directory b_run.ans in the out00001.* file.

Questions:

1. What is the KML and KMT for a 12 ft draft?
2. What is the righting arm at 28 deg roll?
3. At what longitudinal location does the shear force cross the zero axis for the wave crest at location 0.0 ?
4. At what longitudinal location does the shear force cross the zero axis for the wave crest at location 200.0?

Exercise B

The number of stations defined effects the results. In the file b_run.dat change line 20 which now reads

```
plane 0 100 200 300 400 -rect 0 25 100
```

to

```
plane 0 50 100 150 200 250 300 350 400 -rect 0 25 100
```

Delete the b_run.dba directory. Then re–run the analysis. Did the graphs change?

2.2 Basic Stability

Topics:

- Righting arms
- Defining weights
- Model summary reports

Reference files: bstab.cif, bstab.dat, wcomp.cif, wcomp.dat

Discussion

The files bstab.cif and bstab.dat are example files that come with MOSES. They are located in the ultra/hdesk/runs/samples/hystat directory. (This assumes that you installed MOSES with the defaults. If not, the directory is below wherever you installed the ultra directory).

There is a plethora of information on the website. The discussions and exercises hopefully will also provide a good tour through the website.

The discussion for these two files are located at:

- http://bentley.ultramarine.com/hdesk/runs/samples/hystat/bstab.htm
- http://bentley.ultramarine.com/hdesk/runs/samples/hystat/wcomp.htm

After you have read the discussion, see if you can answer the following questions:

Bstab Questions

1. For the Draft 7 ft with a KG = 5.5 ft what is the GML for CBRG180?
2. What is the Y radii of gyration (K–Y) CBRG180?
3. Why are there only ten rows in the RIGHTING ARM RESULTS report?

Change the top of the file to read:

```
&dimen -DIMEN FEET KIPS
&device -oecho no
```

If you look up the command *&device* in the user's manual, you will see that this option controls part of the output.

4 What is different between this output and the original output?

Wcomp Questions

1. What is the maximum amount of ballast for compartment 4C?
2. What compartments are filled to 100%?
3. What is the area ratio at 12.5 degrees?

Exercise

Add the following lines to wcomp.dat:

```
&describe body cbrg180
&describe part cargo
pgen cargo -cs_curr 1 1 1 -cs_win 1 1 1
   plane 50 70 100 130 -rect 14 40 20
end
```

In the hydrostatic section of wcomp.cif, alter so that it reads:

```
HSTATICS
$
$**********************************************      stability trans.
$
   RARM 2.5 10 -WIND 100
      REPORT
   END
   tank_capacity 5p 1
      report
   end
end
```

Questions

1. What is the free surface moment for compartment 5P?
2. What is the area ratio at 12.5 degrees?

2.3 Free Surface Correction

Topics:

- Tank ballast reporting
- Compartment ballasting options

Reference files:

/ultra/hdesk/runs/samples/hystat/fs_mom.cif, fs_mom.dat

Discussion

Let's start by reviewing the DAT file. Here we use the one of the barges from the barge library, SMIT5. There have been three weights added to the barge. The location of the weights is set with the points *wg1, *wg2, and *wg3. Notice that we specified which units we are using with the *&dimen* statement. It is always a good idea to specify the units at the top of the file.

Let's look at the dimension statements in the CIF and the DAT file. This analysis will start using meters and m–tons. The command in the DAT file tells MOSES to:

- Save the current dimensions (-save) then,
- Accept all future input as feet and kips (-dimen feet kips).

The last command in the DAT file (*&dimen -remember*) tells MOSES to return to the previous (saved) dimensions.

All of this might seem elementary, but by using the *–save* and *–remember* options, many errors due to units can be avoided.

For this exercise, we will discuss the CIF file and the output it produces at the same time. It would be a good idea to run the analysis and have the log and output file available.

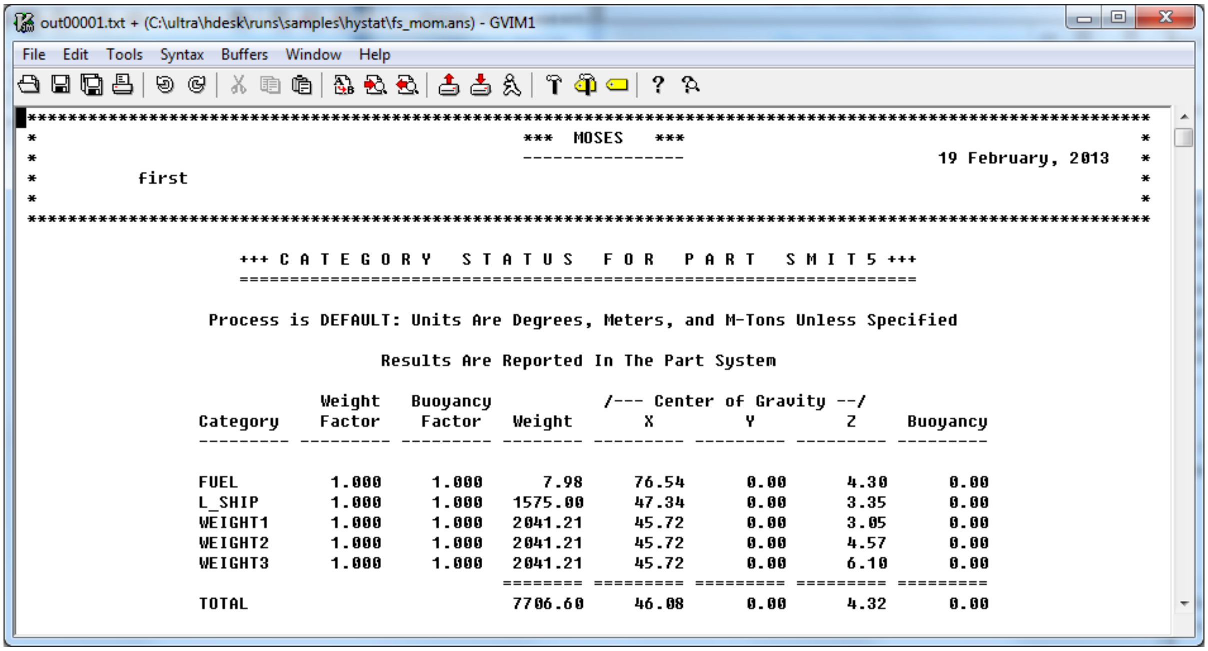
The CIF file has many of the commands that we have seen before. The condition is set by *&instate*, then some ballast water is put in the compartments with *&compartment*. Then, we make a selector named ":tow" to pick six compartments for MOSES to work with. Notice that two of the compartments we chose for MOSES to work with are also compartments that we have placed water in.

A selector name begins with the colon (:) character. Just as in any language, there are special characters in MOSES. In MOSES, the colon is a special character. The complete list of special characters can be found at:

http://bentley.ultramarine.com/hdesk/ref_man/cmd_menu.htm

Next, we ask for a report of the categories with *&status cat*. If we look at the out file, we see that there are five categories that have been defined. There are the three

weights we saw in the DAT file and the barge comes with two categories of its own, FUEL and L_SHIP. (L_SHIP stands for lightship).




```
**********************************************************************************************
*                                   ***  MOSES  ***                                          *
*                                   ---------------                    19 February, 2013     *
*        first                                                                               *
*                                                                                            *
**********************************************************************************************

                 +++ C A T E G O R Y   S T A T U S   F O R   P A R T   S M I T 5 +++
                 ===================================================================

              Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

                                Results Are Reported In The Part System

                         Weight   Buoyancy          /--- Center of Gravity --/
              Category   Factor    Factor   Weight      X         Y        Z      Buoyancy
              --------- --------- --------- -------- --------- --------- --------- ---------

              FUEL        1.000     1.000      7.98    76.54      0.00      4.30      0.00
              L_SHIP      1.000     1.000   1575.00    47.34      0.00      3.35      0.00
              WEIGHT1     1.000     1.000   2041.21    45.72      0.00      3.05      0.00
              WEIGHT2     1.000     1.000   2041.21    45.72      0.00      4.57      0.00
              WEIGHT3     1.000     1.000   2041.21    45.72      0.00      6.10      0.00
                                            ======== ========= ========= ========= =========
              TOTAL                         7706.60    46.08      0.00      4.32      0.00
```

Figure 3: Category Summary

Please notice that the table header reads “Category Status for Selected Parts.” For our analysis, there is only one body and one part so we do not have to worry about getting confused. In later exercises, we will be working with several parts so we need to read the table headers carefully to fully understand their contents.

If you scan the CIF file, you will notice that there are titles added (first, second, third) in lines 17, 22, and 28. In the reports with the title “first,” the barge is not in equilibrium. In the reports with the title “second,” ballast water has been added with the *&cmp_bal* command and the barge is now in equilibrium. The commands *&status b_w –hard* and *&status compart –hard* produce the tables titled “Buoyancy and Weight for SMIT5” and “Compartment Properties” in the output file. For those with the page title “second,” we have not specified a fill type, so the default of “CORRECT” is used. For those with the page title “third,” we have specified the fill type for compartments 1PSS, 3PSC, 3SBC, 5PSS and 5SBS. Specifying the fill type was done with the command

```
&compart -app_none :tow
```

If we read the manual, we find that *–APP_NONE* uses the correct CG when it is filled and uses zero for the derivatives (no free surface correction).

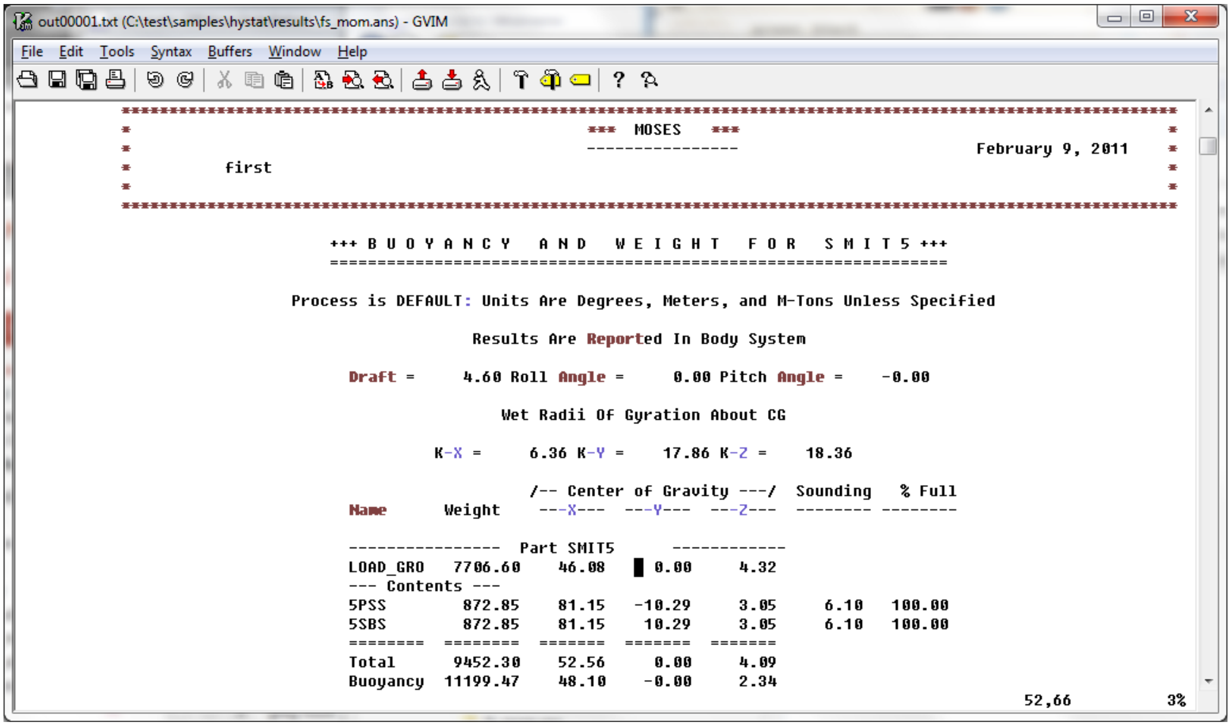
Now if we return to the output we see that

GM is not reported in “first,”

GM is reported as 12.60 meters in “second,”

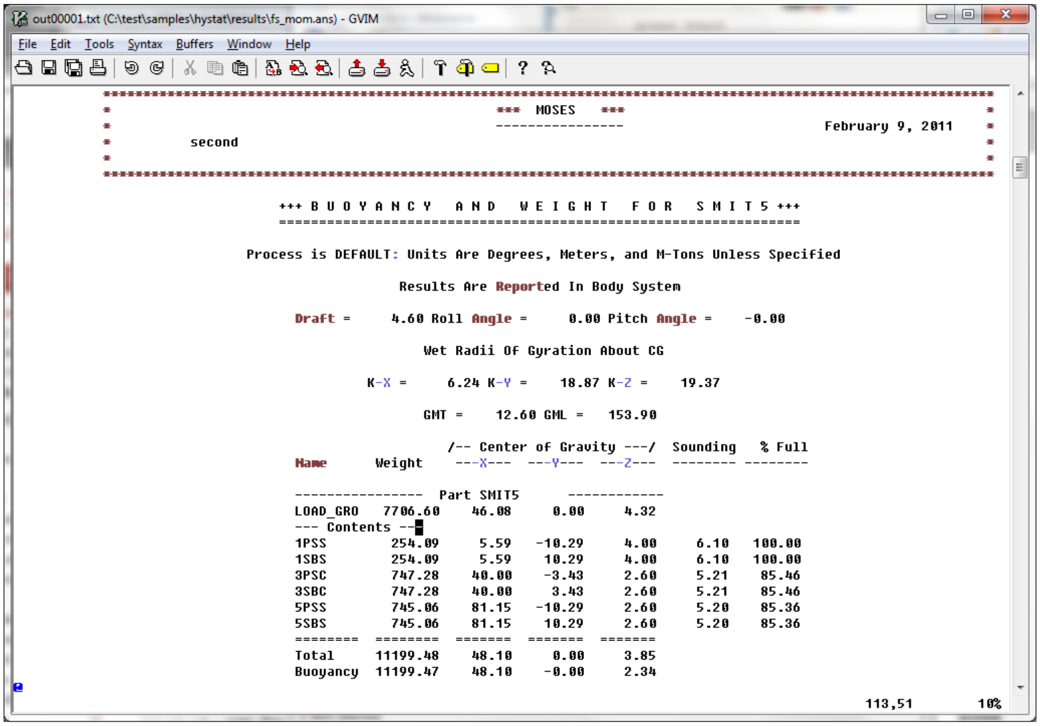
GM is reported as 12.80 meters in “third,”

The three pages from the output are shown below:



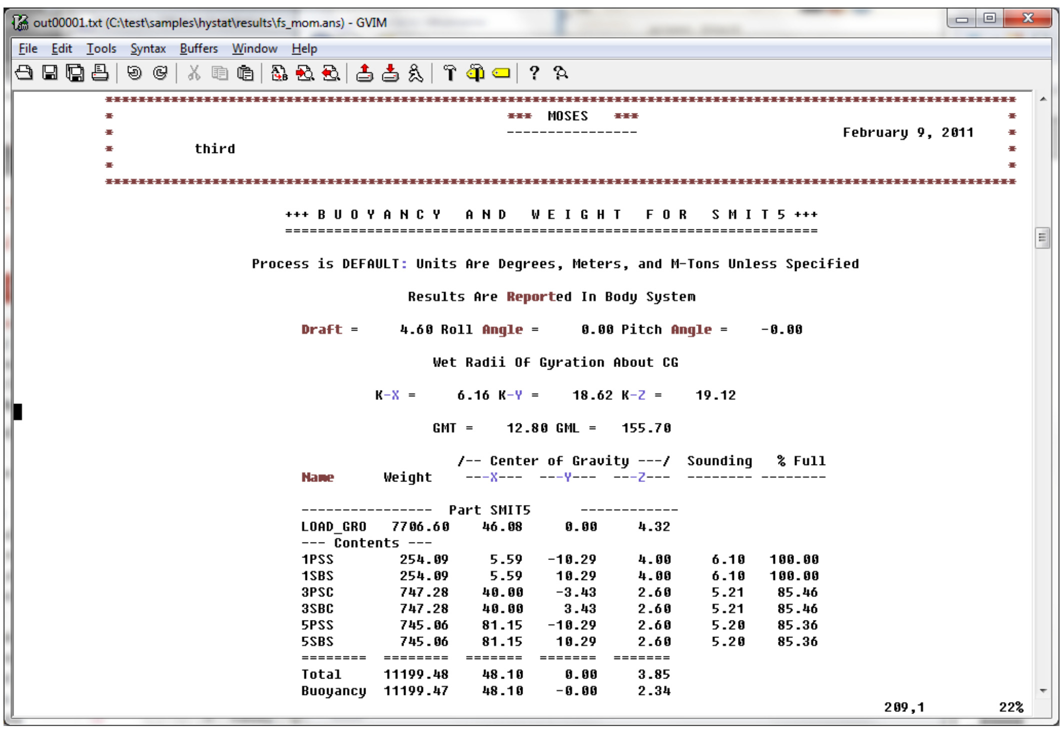

```
*************************************************************************************************
*                                        ***  MOSES  ***                                        *
*                                        ---------------                  February 9, 2011     *
*          first                                                                                *
*                                                                                               *
*************************************************************************************************

                       +++ B U O Y A N C Y   A N D   W E I G H T   F O R   S M I T 5 +++
                       =================================================================

                 Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

                                  Results Are Reported In Body System

                       Draft =      4.60 Roll Angle =     0.00 Pitch Angle =    -0.00

                                     Wet Radii Of Gyration About CG

                               K-X =      6.36 K-Y =     17.86 K-Z =     18.36

                                  /-- Center of Gravity ---/  Sounding   % Full
                       Name       Weight   ---X---  ---Y---  ---Z---  --------  --------

                       ----------------  Part SMIT5     ------------
                       LOAD_GRO  7706.60     46.08     0.00     4.32
                       --- Contents ---
                       5PSS       872.85     81.15   -10.29     3.05      6.10    100.00
                       5SBS       872.85     81.15    10.29     3.05      6.10    100.00
                       ========  ========  =======  =======  =======
                       Total      9452.30     52.56     0.00     4.09
                       Buoyancy  11199.47     48.10    -0.00     2.34
                                                                                 52,66          3%
```

Figure 4: Results of &status b_w when not in equilibrium



```
out00001.txt (C:\test\samples\hystat\results\fs_mom.ans) - GVIM
File Edit Tools Syntax Buffers Window Help

*****************************************************************************************
*                                  ***  MOSES  ***                                      *
*                                  ---------------               February 9, 2011       *
*        second                                                                         *
*                                                                                       *
*****************************************************************************************

                     +++ B U O Y A N C Y   A N D   W E I G H T   F O R   S M I T 5 +++
                     ==================================================================

                Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

                               Results Are Reported In Body System

                      Draft =     4.60 Roll Angle =     0.00 Pitch Angle =    -0.00

                                  Wet Radii Of Gyration About CG

                            K-X =     6.24 K-Y =    18.87 K-Z =    19.37

                                  GMT =    12.60 GML =   153.90

                                 /-- Center of Gravity ---/  Sounding   % Full
                      Name      Weight   ---X---  ---Y---  ---Z---  --------  --------

                      ----------------  Part SMIT5      ------------
                      LOAD_GRO   7706.60    46.08     0.00     4.32
                      --- Contents ---
                      1PSS        254.09     5.59   -10.29     4.00      6.10    100.00
                      1SBS        254.09     5.59    10.29     4.00      6.10    100.00
                      3PSC        747.28    40.00    -3.43     2.60      5.21     85.46
                      3SBC        747.28    40.00     3.43     2.60      5.21     85.46
                      5PSS        745.06    81.15   -10.29     2.60      5.20     85.36
                      5SBS        745.06    81.15    10.29     2.60      5.20     85.36
                      ========  ========  =======  =======  =======
                      Total      11199.48    48.10     0.00     3.85
                      Buoyancy   11199.47    48.10    -0.00     2.34
                                                                          113,51        10%
```

Figure 5: Results of &status b_w with type “correct”



```
out00001.txt (C:\test\samples\hystat\results\fs_mom.ans) - GVIM
File Edit Tools Syntax Buffers Window Help

*****************************************************************************************
*                                  ***  MOSES  ***                                      *
*                                  ---------------               February 9, 2011       *
*        third                                                                          *
*                                                                                       *
*****************************************************************************************

                     +++ B U O Y A N C Y   A N D   W E I G H T   F O R   S M I T 5 +++
                     ==================================================================

                Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

                               Results Are Reported In Body System

                      Draft =     4.60 Roll Angle =     0.00 Pitch Angle =    -0.00

                                  Wet Radii Of Gyration About CG

                            K-X =     6.16 K-Y =    18.62 K-Z =    19.12

                                  GMT =    12.80 GML =   155.70

                                 /-- Center of Gravity ---/  Sounding   % Full
                      Name      Weight   ---X---  ---Y---  ---Z---  --------  --------

                      ----------------  Part SMIT5      ------------
                      LOAD_GRO   7706.60    46.08     0.00     4.32
                      --- Contents ---
                      1PSS        254.09     5.59   -10.29     4.00      6.10    100.00
                      1SBS        254.09     5.59    10.29     4.00      6.10    100.00
                      3PSC        747.28    40.00    -3.43     2.60      5.21     85.46
                      3SBC        747.28    40.00     3.43     2.60      5.21     85.46
                      5PSS        745.06    81.15   -10.29     2.60      5.20     85.36
                      5SBS        745.06    81.15    10.29     2.60      5.20     85.36
                      ========  ========  =======  =======  =======
                      Total      11199.48    48.10     0.00     3.85
                      Buoyancy   11199.47    48.10    -0.00     2.34
                                                                          209,1         22%
```

Figure 5: Results of &status b_w with type “app_none”

Let’s concentrate on the GM change between the second to the third case. If you remember from your naval architecture class, free surface in the compartments reduces the metacentric height. What is being shown here is that the third set in which

we specified the compartments to with a fill type of *APP_NONE* do not have a free surface moment correction.

Let's look briefly at the commands towards the end of the file. We have used the *cform* command before to generate the curves of form. The curves of form include the displacement and the distance from the keel to the metacenter (KM). Note that the curves of form are based strictly on the hull geometry.

Also, toward the end of the command file we have *tank_capacity*. This command generates the tables titled "Tank Capacities for XXXX" where the XXXX is the compartment name. The columns we are interested in here are the Free Surf. Moment – Trans. and Long. If we review the Buoyancy and Weight report, we see that the 3XXX and the 5XXX compartments are filled to 85%. The Tank Capacities report the free surface moment to be 562 for the 3XXX and 561 for the 5XXX. We will work with 562 for the calculations here.

Calculating the free surface is simple enough. We should be able to verify the change of 0.2 meters. We know that Free Surface Correction is the Free Surface Moment divided by Displacement.

$$fs\_cor = fs\_mom/displacement \qquad (2.3 \qquad 1)$$

So far, we have the free surface moment. For the displacement, we look at the buoyancy and weight report to get the draft of 4.60 meters or we can look at the curves of form for the displacement 11199.47 m–tons. This would be 10,929 m^3. If we do the math, we find that each compartment reduces the metacentric height by 0.05 meters. Since we have four compartments, the metacentric height is reduced by 0.2 meters. And we see this is true. The difference between the GM with (12.60 m) and without (12.80 m) free surface correction is 0.2 meters.

What else does this influence? The free surface correction also influences the righting arms. Here the righting arms are reported and plotted. The reports titled "back to correct" include the free surface correction. As you can see, the righting arm without the free surface correction has a maximum of 2.41 around 24 degrees. Whereas the righting arm with the free surface correction has a maximum of 2.33 around 24 degrees.

Exercise A

Change line number 9 to read:
`&dimen -dimen feet kips`

Take out the *&dimen -rem* line in the data file.

Rerun the analysis and see what changed.

Exercise B

In the previous exercise, you turned on the wind model with the option *–cs_cur* and *–cs_win*. That was for a barge that we modeled just for the exercise. Here we are using a barge from the barge library. Go to:

http://bentley.ultramarine.com/hdesk/tools/vessels/vessels.htm

And read how making the following alteration to the DAT file

```
&set v_cur = 1
&set v_win = 1
```

will turn on the wind and current part of the model.

1. What is the wind heeling area at 38 degrees?

2.4 Compartment Ballasting using Valves

Topics:

This exercise is a continuation of how to work with compartments. Here the draft of a barge is changed three times. The barge has three compartments and each compartment has a valve on the outside shell at a different height. Once the valves enter the water, water is allowed to enter the compartment. When the barge draft is decreased the water inside the compartment, up to the height of the valve, is trapped inside. This will be the first exercise where we use sensors and macros.

Reference files:

/ultra/hdesk/runs/samples/hystat/static_open.cif, static_open.dat

Discussion

For this analysis a good part of the work is done in perparation as part of the data file. Let's start by reviewing the DAT file. Here we use a very shoe box looking barge. The section that defines the barge is the following.

```
&describe body test
pgen test -perm 1
   plane 0 20 -rect 0 6 10
end_pgen
```

The commane *&describe body* test creates a body, part, compartment and piece with the name of "test". MOSES automatically creates an association to the part, compartment, and piece for every body. For now we are happy to know that there is a body with the name given, and that an association with the part, compartment, and piece has been made.

The command *pgen* is short for "piece generator". The command *&describe piece test* could have also been used before the *pgen* command. It is not necessary to have the *&describe piece* followed by the *pgen* command. The *pgen* command and the *end_pgen* will be sufficient. The option *–perm 1* tells MOSES it is an outer shell, therefore it should be included in any displacement calculations. The number following *–perm* is used as a multiplier for the calculated displacement. In this case MOSES is to multiply the calculated displacement by 1.

This is the first time we will be defining compartments with flood and vent valves. The points that will be used to locate the valves are part of the body. The naming of the points that will be used to locate the valves is part of the body description. Here is what section looks like.

```
*C1 5 10 2
*V1 5 10 20
*C2 10 10 3
```

```
*V2 10 10 20
*C3 15 10 5
*V3 15 10 20
```

We need to first define the points before we use them in the compartment description. This file is also heavily commented. Each of the point defintions is followed by a short description of how it will be used. It is a good idea to try to pick names that will make the output easier to use. Here each flood valve has a point with the letter "C" and each vent valve has a point with the letter "V". This is just for convinence.

In the next sections the compartments along with the valves are defined. The same format is taken for each compartment, so this discussion will focus only on the first.

```
&describe hole v1 v_valve -area 3.14158*5625 -point *V1
&describe hole hc1 f_valve -area 3.14158*5625 -point *C1
&describe compartment C1 -holes Hc1 V1
pgen -perm -0.95 -diftype none
   plane 0 10 -rect 0 6 6
end_pgen
```

The first two lines associate the points with a valve. The manual reference for this command is at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/cmp_int.htm#&DESCRIBE HOLE

The first step is to define each hole and associate it with its intended use. For the vent valve we are naming it "V1" and have used the type designation of "v_valve". The two options *–area* and *–point* tell MOSES what area and location to use. These are options and if we did not use them MOSES has defaults it would use. However, the location of the valve is important to our analysis. It would be acceptable to leave the *–area* option out.

This is the first time we will be defining a compartment. The general format is similar to that of defining the outer shell. Different options are used when we define the compartment. The holes are associated with the compartment as part of the *&describe compartment* command. As part of the *pgen* command we assign a permiability with the option *–perm* and designate its diffraction type as "none" with the option *–diftype*. The rest of the definition should be a review of the *pgen* command used earlier.

In the last of the model definition begins again with *&describe body test*. In this section we are defining sensors at the valve locations. One sensor for each valve. The manual reference page for sensor definition is at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/sensor.htm#&DESCRIBE SENSOR

The sensor are named with the letter "s" being added to the point name. Just as

with the hole defintion the point is associate with each sensor. Here we use the option *–signal s_type s_source s_desired s_val s_b s_n*. And the option *–limits lim_l lim_u*.

For the first option *–signal* the following is how the values are interpretted.

s_type = point
s_source = ∗c1
s_desired =
s_val = value
s_b = 3rd value
s_n =

The tricky part may be figuring out that s_desired was left blank. An acceptable input for s_desired has to be a numeric value, when MOSES reads a word as the third entry it knows that s_desired is not being set. Instead it sees that the word "value" is an acceptable input for s_val.

For the second option *–limits* it is easy to see the lower limit is -0.1 and the upper limit is 100. This means that any time the third value of the point location is between -0.01 and 100 the sensor value will be ".false.".

This section is repeasted for compartment C2 and C3. Then a selector is defined.

```
&select :hole -select *C1 *C2 *C3
```

Selectors are a means to group a set of entities. Here we are anticipating needing information about the three points. It will be much easier to ask for the information for the group with the :hole name, instead of typing out all three points.

In the next section we define a macro. Macros are used to create new commands. There are many analysis where several configurations are examined and the same set of data is reported. If a command that creates the desired output is used this saves us from human error. The command that we will be creating is "checkloc".

The purpose of this macro is do three things:

1. to report the valve z location (3rd value of the point location),
2. report the value of the sensor, and
3. if the sensor is .true., open the valve.

The same sequence is done three times with a name change. We are going to review

line by line only for one set.

The first line:

```
&type **********
```

is meant to draw our attention.

The second line

```
&type alarm c1 &info(alarm_sensor sc1)
```
reports the value of the sensor. This will be either ".false." or ".true.".

The third line reports the global coordinate location of the point.

```
&type *C1 &point(coord *C1 -g)
```

The next set of commands are an if statement, therefore, we are going to treat as a group.

```
&if &info(alarm_sensor sc1) &then
   &compartment -open_valve c1 -percent c1 100
&endif
```

From two lines above we can see what the value of *&info(alarm_sensor sc1)* is. We will be able to see if a ".false." or a ".true" we substituted for the middle section of this command. If a ".false." is substituted then nothing happens.

If a ".true." is substituted then, the command *&compartment –open_valve c1 -percent c1 100* `is executed.`

The manual page for this command can be found at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/cmp_fill.htm#&COMPARTMENT

In order for a change in ballast water inside the compartment to occur we first have to tell MOSES the maximum volume of the fluid in the compartment can be. The manual page states:

> The change of fluid in the compartment occurs statically, and can be observed using &status compartment. The maximum volume of fluid in the compartment is artificailly limited by that specified using the add ballast options.

This is why after the *–open_valve* option we find the *–percent* option. the *–percent* option is telling MOSES the maximum volume limit of fluid in the compartment.

This sequence of commands is done for each of the valves. The last command *&end-*

macro tells MOSES the macro definition has ended. The data file ends.

Discussion: Command File

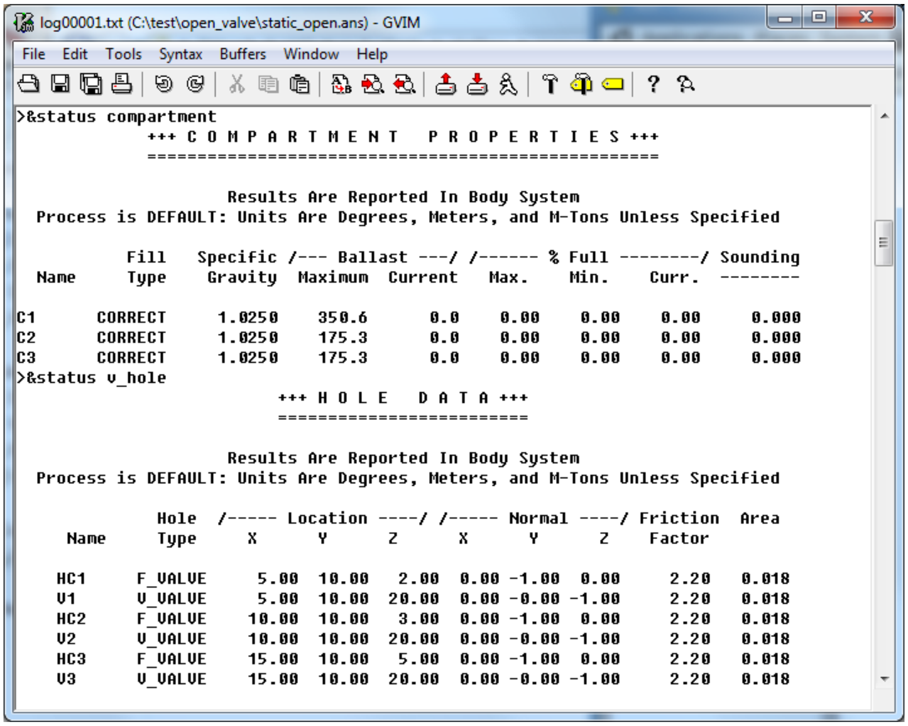
The command file begins as many of the other files have. It sets the units for the analysis, *&dimen -dimen meters m-tons*. This file shows how to define the specific gravity of water with *&default –spgwater 1.025*. Then we tell MOSES we do not care to see an echo of the input model as part of the output file. And finally we use the command *inmodel* to read in the model.

The command file is set up in sections. Each section is assigned an event number with the command *&event_store*. After the *inmodel* the event number is 1. We use the macro "checkloc" to check that location of the valves and the value of the sensors. The following is what you should see in the log file as a result of the macro "checkloc".

```
****************
alarm c1 .FALSE.
*C1 5 10 2
****************
alarm c2 .FALSE.
*C2 10 10 3
****************
alarm c3 .FALSE.
*C3 15 10 5
```

We see that this is the location defined in the data file. This makes sense since after the inmodel the body coordinate system origin is at the global coordinate system origin.

The next three commands shows further detail of the buoyancy and weight (b_w), the compartments, and the valve hole data (v_hole). Remember the manual did say we could observe the resulting change in ballast with the *&status compartment* command. The results of these *&status* commands, confirm that the compartments are empty and the valve are closed.



```
>&status compartment
             +++ C O M P A R T M E N T   P R O P E R T I E S +++
             ===================================================

                     Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

           Fill    Specific /--- Ballast ---/ /------ % Full --------/ Sounding
  Name     Type     Gravity  Maximum  Current   Max.    Min.    Curr.  --------

C1      CORRECT     1.0250    350.6      0.0    0.00    0.00    0.00     0.000
C2      CORRECT     1.0250    175.3      0.0    0.00    0.00    0.00     0.000
C3      CORRECT     1.0250    175.3      0.0    0.00    0.00    0.00     0.000
>&status v_hole
                          +++ H O L E   D A T A +++
                          =========================

                     Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

              Hole   /----- Location ----/ /----- Normal ----/ Friction  Area
     Name     Type      X       Y       Z      X      Y      Z    Factor

    HC1     F_VALVE     5.00  10.00    2.00  0.00 -1.00  0.00     2.20   0.018
    V1      V_VALVE     5.00  10.00   20.00  0.00 -0.00 -1.00     2.20   0.018
    HC2     F_VALVE    10.00  10.00    3.00  0.00 -1.00  0.00     2.20   0.018
    V2      V_VALVE    10.00  10.00   20.00  0.00 -0.00 -1.00     2.20   0.018
    HC3     F_VALVE    15.00  10.00    5.00  0.00 -1.00  0.00     2.20   0.018
    V3      V_VALVE    15.00  10.00   20.00  0.00 -0.00 -1.00     2.20   0.018
```


Figure 6: Results of &status for Event 1

Three figures are generated to get an understanding of the setup.

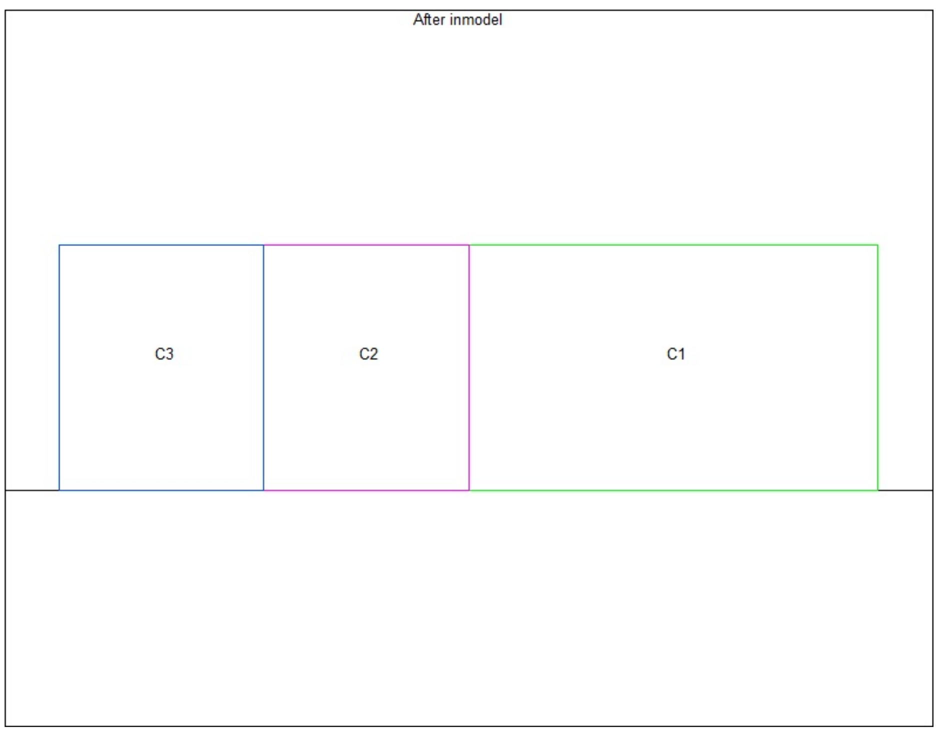


Figure 7: Compartment designation

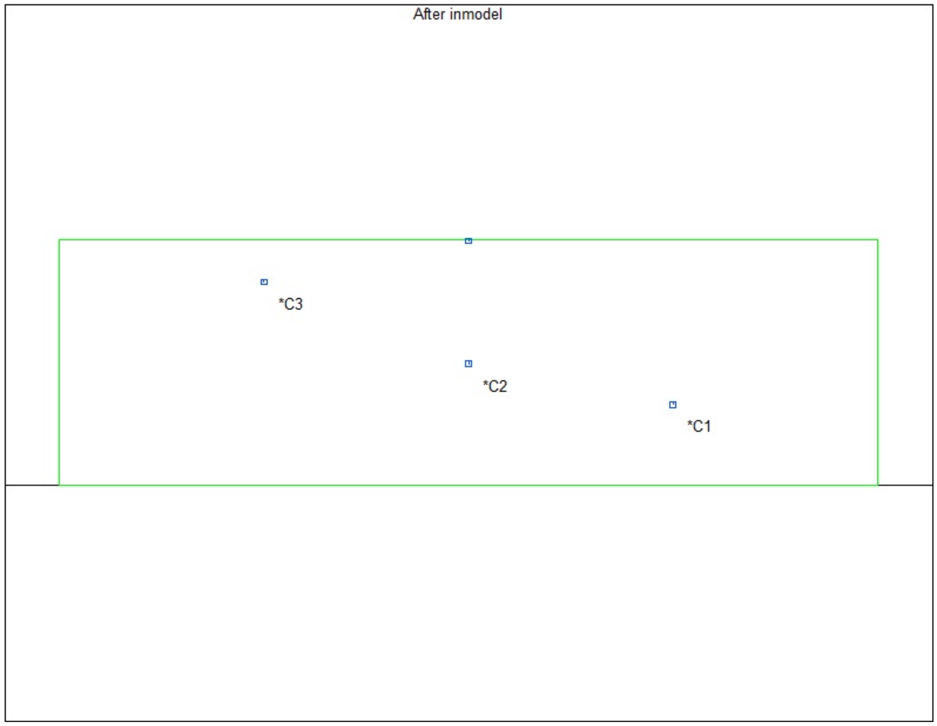


Figure 8: Default view with valve locations

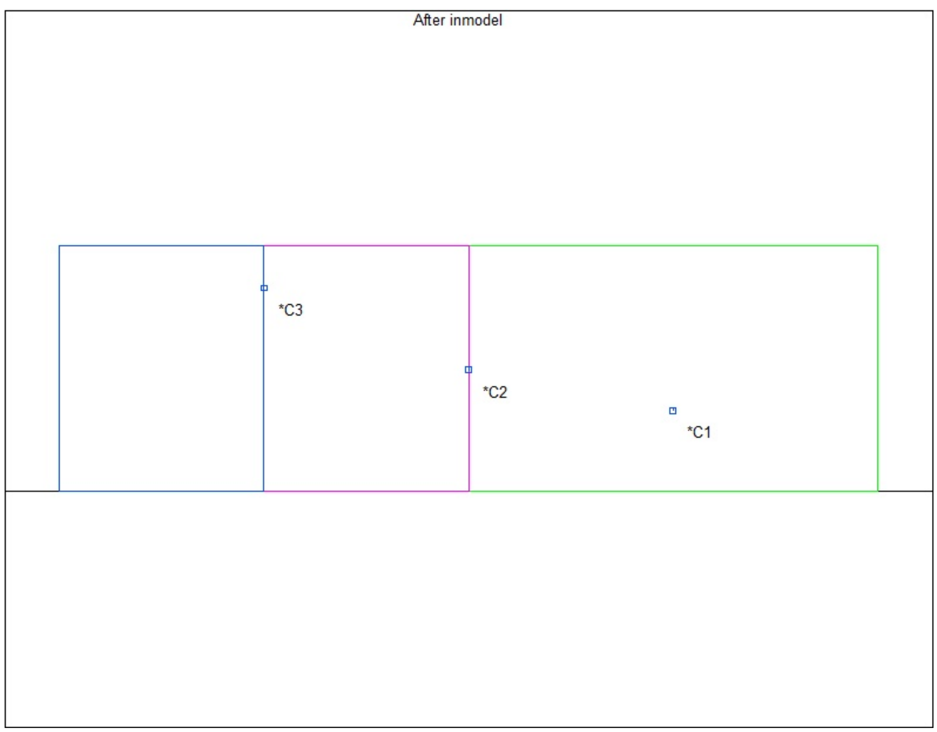


Figure 9: Compartments with valve locations

This gets us the comment that reads "Filling up C1 with sounding 2.2m". We have seen some of the commands in this section. The title is set with *&title Draft 2.2m*.

Then the condition is changed with *&instate tests –locate 0 0 -2.2 0 0 0* The condition will be stored as event 2. And we use our "checkloc" macro. In the log file we find:

```
****************
alarm c1 .TRUE.
*C1 5 10 -0.2000001
****************
alarm c2 .FALSE.
*C2 10 10 0.7999998
****************
alarm c3 .FALSE.
*C3 15 10 2.8
```

We see that the z values of the locations has changed. This makes sense, since we changed the draft with the *&instate –locate* command. Since the valve in compartment C1 was opened we need to check how much water is now in the compartment. We need to review the results of the *&status* commands.

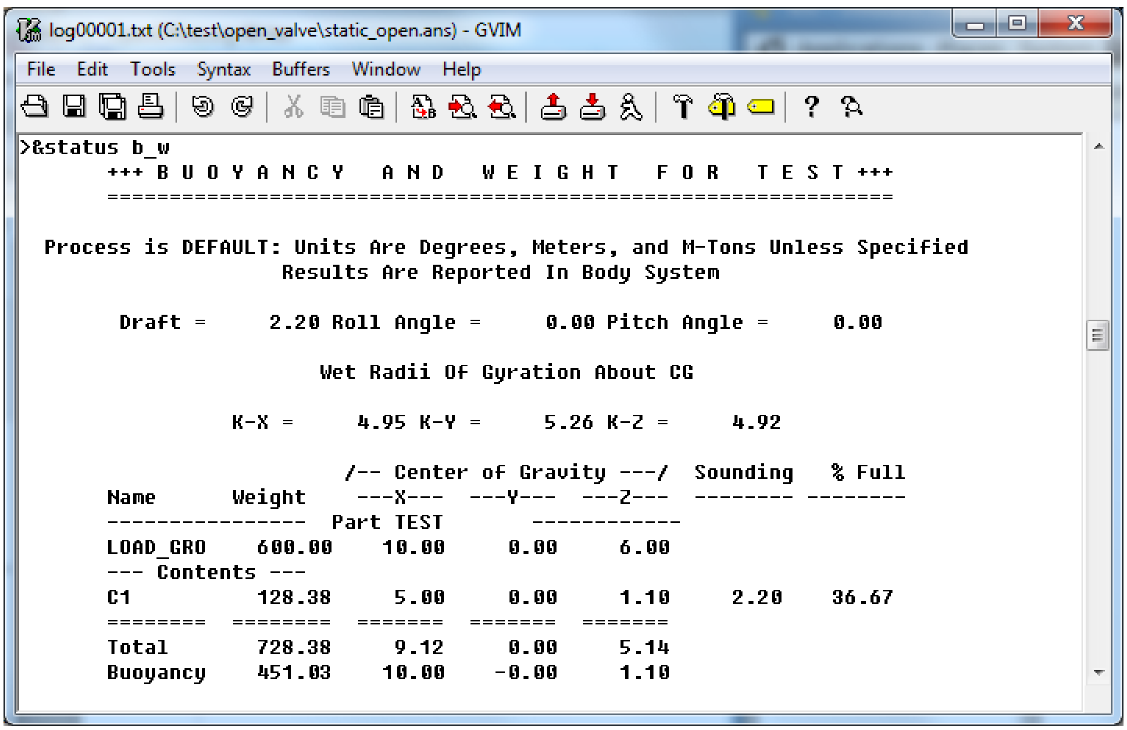
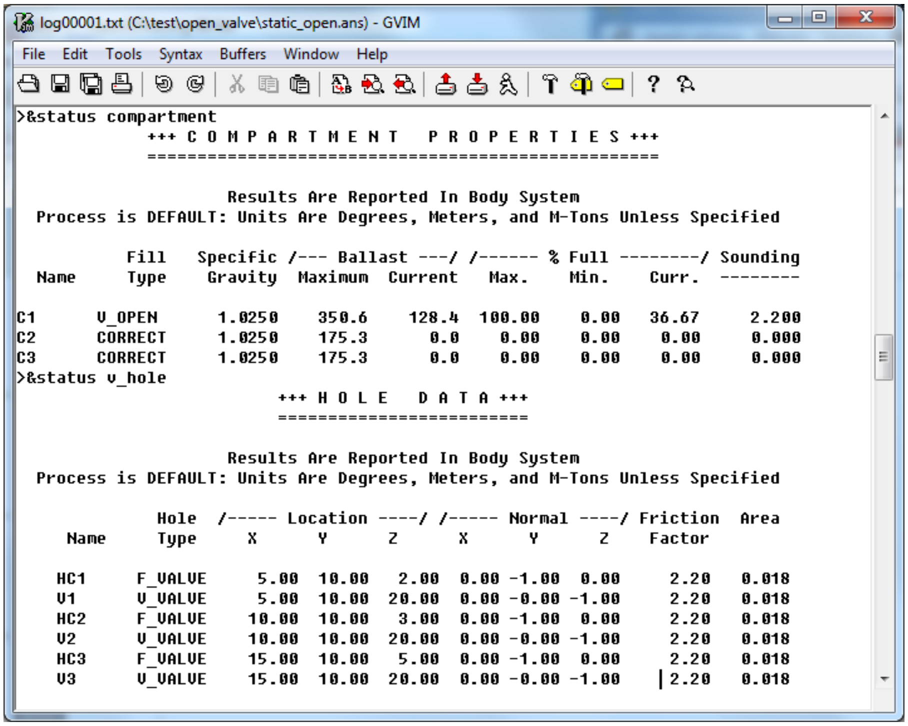


Figure 10: Buoyancy and Weight Report for Event 2




```
>&status compartment
               +++ C O M P A R T M E N T   P R O P E R T I E S +++
               ===================================================

                       Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

           Fill    Specific /--- Ballast ---/ /------ % Full --------/ Sounding
  Name     Type     Gravity  Maximum  Current   Max.    Min.    Curr.  --------

C1      V_OPEN      1.0250    350.6    128.4  100.00    0.00   36.67     2.200
C2      CORRECT     1.0250    175.3      0.0    0.00    0.00    0.00     0.000
C3      CORRECT     1.0250    175.3      0.0    0.00    0.00    0.00     0.000
>&status v_hole
                            +++ H O L E   D A T A +++
                            =========================

                       Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

              Hole  /----- Location ----/ /----- Normal ----/ Friction  Area
     Name     Type      X       Y       Z      X      Y      Z    Factor

    HC1     F_VALVE    5.00  10.00   2.00  0.00 -1.00  0.00     2.20   0.018
    V1      V_VALVE    5.00  10.00  20.00  0.00 -0.00 -1.00     2.20   0.018
    HC2     F_VALVE   10.00  10.00   3.00  0.00 -1.00  0.00     2.20   0.018
    V2      V_VALVE   10.00  10.00  20.00  0.00 -0.00 -1.00     2.20   0.018
    HC3     F_VALVE   15.00  10.00   5.00  0.00 -1.00  0.00     2.20   0.018
    V3      V_VALVE   15.00  10.00  20.00  0.00 -0.00 -1.00     2.20   0.018
```

Figure 11: Compartments Reports for Event 2

We see changes in the buoyancy and weight report and in the compartment properties report. The weight and the sounding of the ballast is included in both reports. In the compartment properties report the value for fill type, ballast maximum and %full maximum has changed. Remember this is establishing the artificial limit set with the macro. Here again is the quote from the reference manual.

> The change of fluid in the compartmnet occurs statically, and can be observed using &status compartment. The maximum volume of fluid in the compartment is artificailly limited by that specified using the add ballast options.

The values in the "Hole Data" report do not change. This report is in the body system. We did not expect a change, becuase we changed the location of the body with reference to the global, not the body. The locations of the valves are not changing, nor are the areas or the friction factors.

The last thing we do as part of this event is make a picture. This picture shows that the mean water level is above valve V1 assoicated with point ∗C1. With the valve open and the water level above the valve location, the water level is the same inside and outside the compartment.

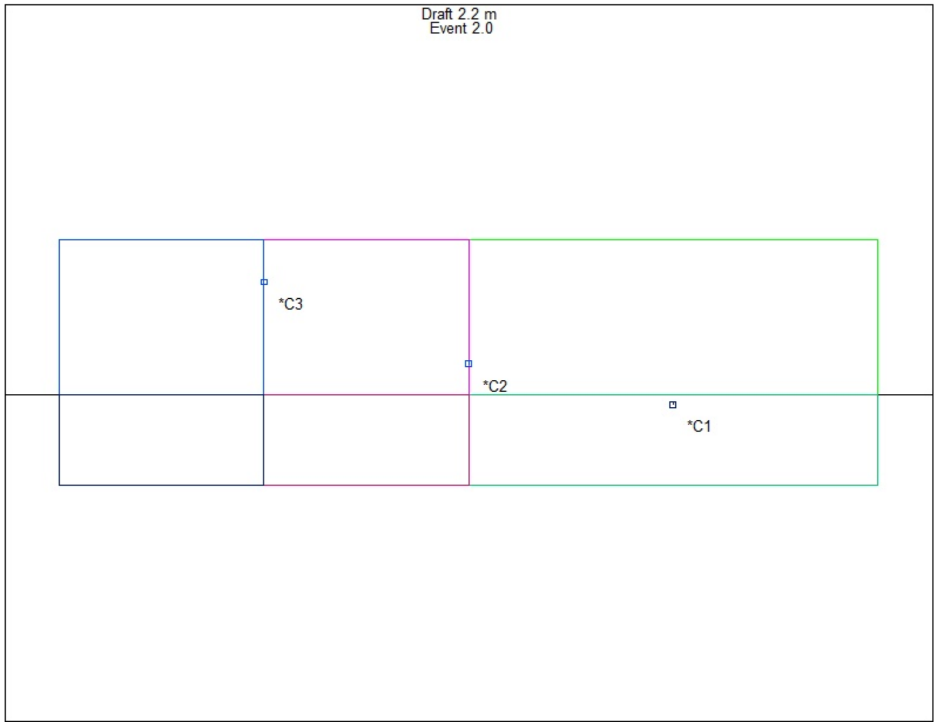
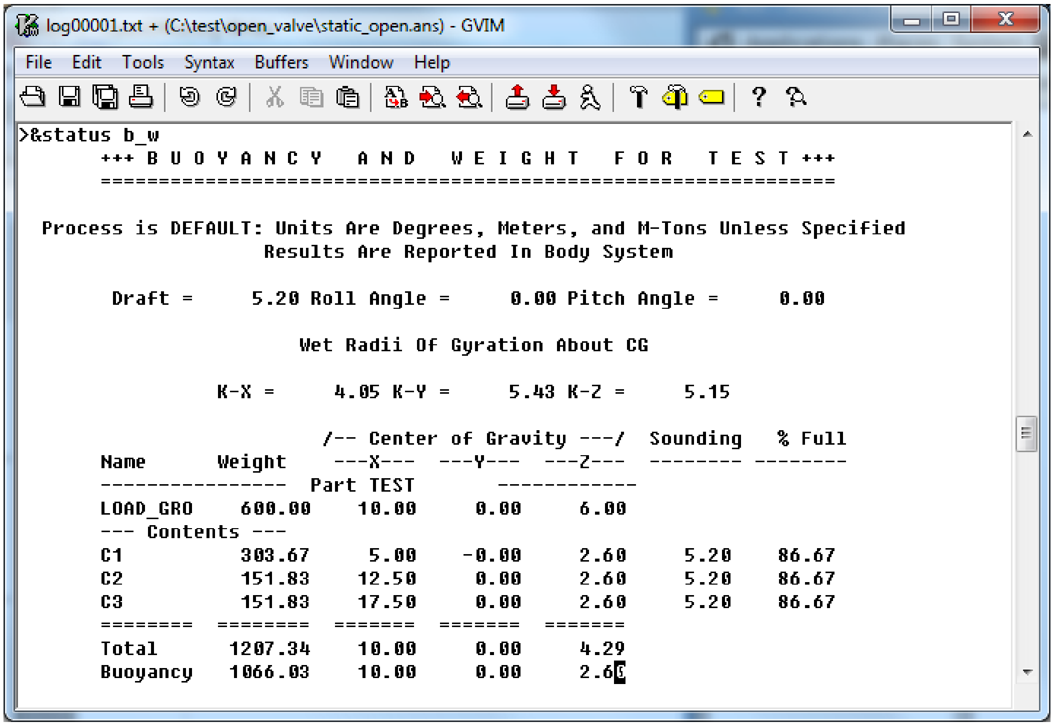


Figure 12: Compartments Reports for Event 2

This brings us to event 3. The same set of commands are used but with different values. Here we are setting the draft for 5.2m. Which means that all three valves will be below the water level. Here are the results of the "checkloc" macro.

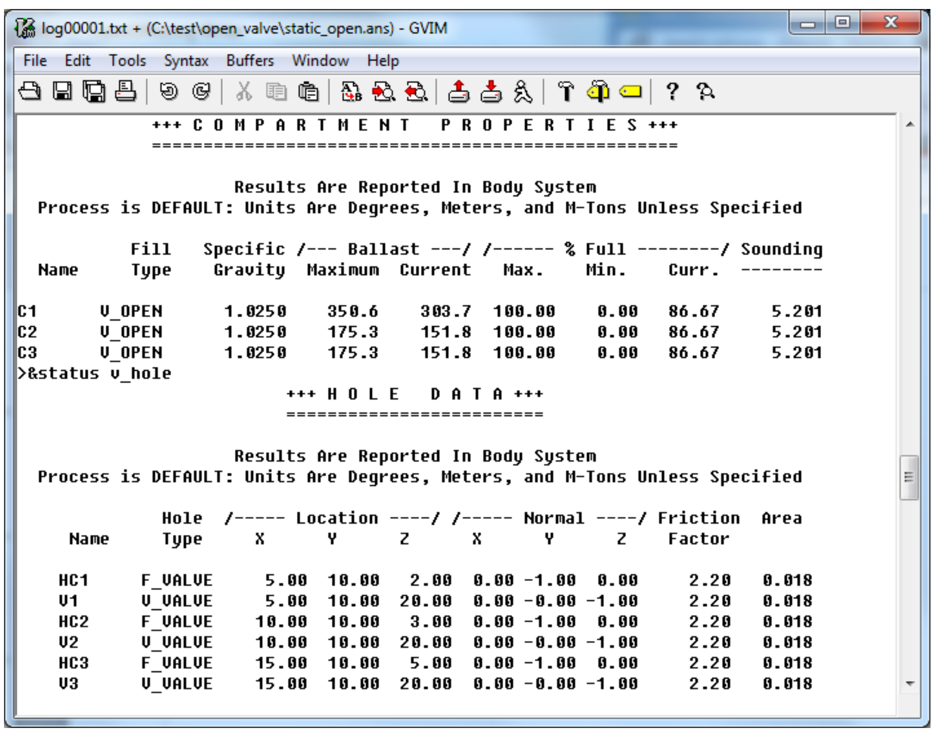
```
****************
alarm c1 .TRUE.
*C1 5 10 -3.2
****************
alarm c2 .FALSE.
*C2 10 10 -2.2
****************
alarm c3 .FALSE.
*C3 15 10 -0.1999997
```

Here is the reports in the log files. We see that all three compartment have a sounding of 5.2m and each corresponds to an 86.67% full. For compartment C1 which is larger, the weight of the ballast is also larger.




```
>&status b_w
      +++ B U O Y A N C Y   A N D   W E I G H T   F O R   T E S T +++
      ===============================================================

  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified
                     Results Are Reported In Body System

       Draft =      5.20 Roll Angle =      0.00 Pitch Angle =      0.00

                        Wet Radii Of Gyration About CG

                K-X =      4.05 K-Y =      5.43 K-Z =      5.15

                       /-- Center of Gravity ---/  Sounding   % Full
      Name      Weight   ---X---  ---Y---  ---Z---  -------- --------
      ---------------- Part TEST          ------------
      LOAD_GRO    600.00    10.00     0.00     6.00
      --- Contents ---
      C1          303.67     5.00    -0.00     2.60     5.20    86.67
      C2          151.83    12.50     0.00     2.60     5.20    86.67
      C3          151.83    17.50     0.00     2.60     5.20    86.67
      ======== ======== ======= ======= =======
      Total      1207.34    10.00     0.00     4.29
      Buoyancy   1066.03    10.00     0.00     2.60
```

Figure 13: Buoyancy and Weight Report for Event 3




```
             +++ C O M P A R T M E N T   P R O P E R T I E S +++
             ===================================================

                    Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

            Fill   Specific /--- Ballast ---/ /------ % Full --------/ Sounding
  Name      Type    Gravity  Maximum  Current   Max.    Min.    Curr.  --------

C1       V_OPEN     1.0250    350.6    303.7  100.00    0.00   86.67     5.201
C2       V_OPEN     1.0250    175.3    151.8  100.00    0.00   86.67     5.201
C3       V_OPEN     1.0250    175.3    151.8  100.00    0.00   86.67     5.201
>&status v_hole
                          +++ H O L E   D A T A +++
                          =========================

                    Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

              Hole  /----- Location ----/ /----- Normal ----/ Friction  Area
     Name     Type      X       Y       Z      X      Y      Z   Factor

    HC1     F_VALVE     5.00   10.00    2.00   0.00 -1.00  0.00     2.20  0.018
    V1      V_VALVE     5.00   10.00   20.00   0.00 -0.00 -1.00     2.20  0.018
    HC2     F_VALVE    10.00   10.00    3.00   0.00 -1.00  0.00     2.20  0.018
    V2      V_VALVE    10.00   10.00   20.00   0.00 -0.00 -1.00     2.20  0.018
    HC3     F_VALVE    15.00   10.00    5.00   0.00 -1.00  0.00     2.20  0.018
    V3      V_VALVE    15.00   10.00   20.00   0.00 -0.00 -1.00     2.20  0.018
```

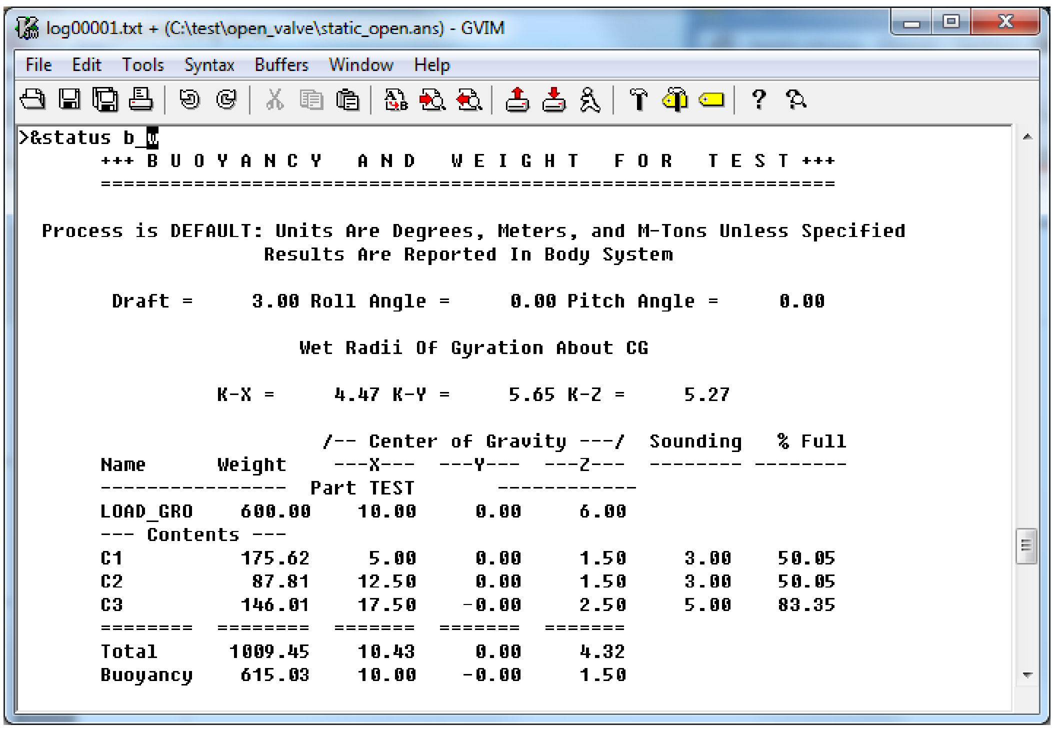
Figure 14: Compartments Reports for Event 3

This brings us to event 4. The same set of commands are used but with different values. This time we are decreasing the draft. This means that when the valve exits

the water, any ballast still in the compartment is trapped. For event 4, the water level for compartments with the valve above the water level should have ballast up to the valve.

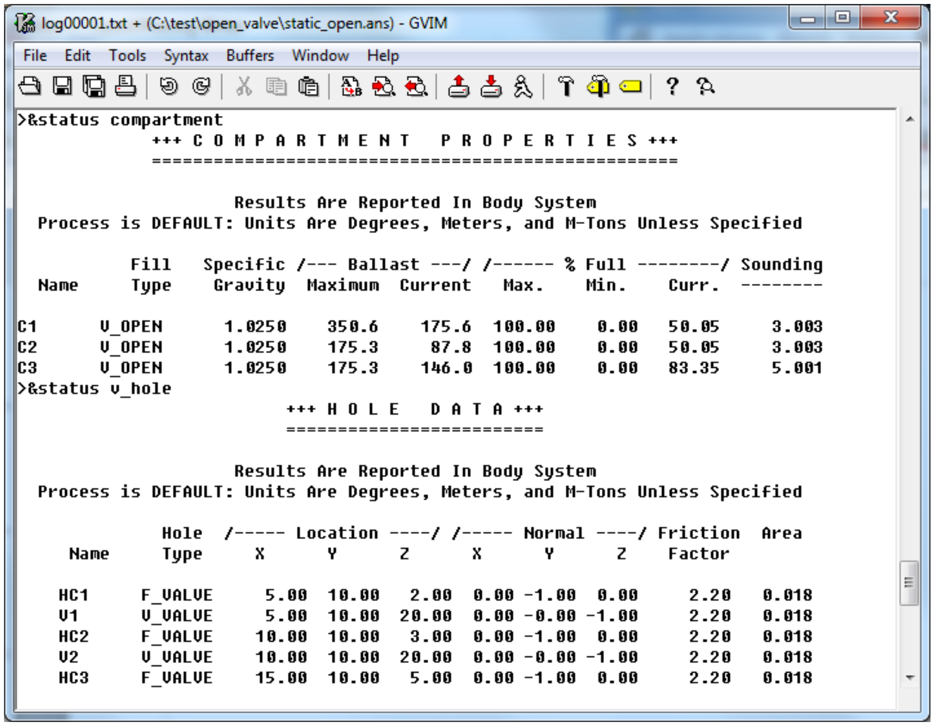
```
* * * * * * * * * * * * * * **
alarm c1 .TRUE.
*C1 5 10 -1
* * * * * * * * * * * * * * **
alarm c2 .FALSE.
*C2 10 10 0
* * * * * * * * * * * * * * **
alarm c3 .FALSE.
*C3 15 10 2
```

Here are the reports in the log file. For compartment C3 the water level is at the valve level. For compartments C2 and C1 the water level outside the compartment is the same as inside the compartmnet.




```
>&status b_w
       +++ B U O Y A N C Y   A N D   W E I G H T   F O R   T E S T +++
       ===============================================================

  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified
                     Results Are Reported In Body System

        Draft =     3.00 Roll Angle =     0.00 Pitch Angle =     0.00

                        Wet Radii Of Gyration About CG

                 K-X =     4.47 K-Y =     5.65 K-Z =     5.27

                         /-- Center of Gravity ---/  Sounding   % Full
       Name      Weight   ---X---  ---Y---  ---Z---  --------  --------
       ----------------  Part TEST         ------------
       LOAD_GRO    600.00    10.00     0.00     6.00
       --- Contents ---
       C1          175.62     5.00     0.00     1.50     3.00     50.05
       C2           87.81    12.50     0.00     1.50     3.00     50.05
       C3          146.01    17.50    -0.00     2.50     5.00     83.35
       ========  ========  =======  =======  =======
       Total      1009.45    10.43     0.00     4.32
       Buoyancy    615.03    10.00    -0.00     1.50
```

Figure 15: Buoyancy and Weight Report for Event 4



```
>&status compartment
             +++ C O M P A R T M E N T   P R O P E R T I E S +++
             ===================================================

                     Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

           Fill    Specific /--- Ballast ---/ /------ % Full --------/ Sounding
  Name     Type     Gravity  Maximum  Current   Max.    Min.    Curr.  --------

C1       V_OPEN      1.0250    350.6    175.6  100.00    0.00   50.05     3.003
C2       V_OPEN      1.0250    175.3     87.8  100.00    0.00   50.05     3.003
C3       V_OPEN      1.0250    175.3    146.0  100.00    0.00   83.35     5.001
>&status v_hole
                          +++ H O L E   D A T A +++
                          =========================

                     Results Are Reported In Body System
  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

              Hole  /----- Location ----/ /----- Normal ----/ Friction  Area
     Name     Type     X      Y      Z      X      Y      Z    Factor

    HC1     F_VALVE    5.00  10.00   2.00  0.00 -1.00  0.00     2.20   0.018
    V1      V_VALVE    5.00  10.00  20.00  0.00 -0.00 -1.00     2.20   0.018
    HC2     F_VALVE   10.00  10.00   3.00  0.00 -1.00  0.00     2.20   0.018
    V2      V_VALVE   10.00  10.00  20.00  0.00 -0.00 -1.00     2.20   0.018
    HC3     F_VALVE   15.00  10.00   5.00  0.00 -1.00  0.00     2.20   0.018
```

Figure 16: Compartments Reports for Event 4

Please notice that for the fill type in the compartment properties report the valve is still open. We did not close the valve, we just relocated the body so that the valve would be above the water line. Here is the picture showing the water level and the location of the valves.

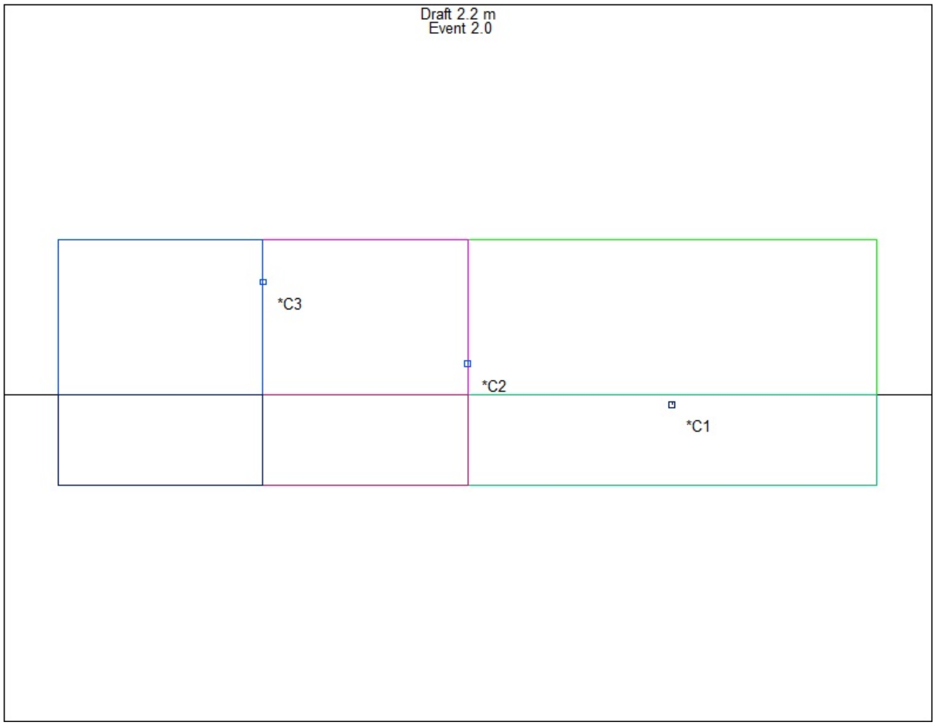


Figure 17: Compartments Reports for Event 2

This concludes the exercise on opening valves in a static analysis.

Questions

During the data discussion I mentioned that the area of the valve is not needed for this analysis. Comment out the valve areas and see if the results change.

2.5 Stability Check and KG Allow

Topics:

- Introduction to Automated Installation Tools
- Working with the stability macros
- Familiarization with the &status command

Reference files: m_stab.cif, m_stab.dat, i_stab.cif, i_stab.dat, box.dat

Discussion

The discussion for i_stab.cif and i_stab.dat are located at:

http://bentley.ultramarine.com/hdesk/runs/samples/hystat/i_stab.htm
http://bentley.ultramarine.com/hdesk/ref_man/install.htm

The discussion for m_stab.cif is located at:

http://bentley.ultramarine.com/hdesk/runs/samples/hystat/m_stab.htm

The file i_stab uses the automatic installation tools to perform a stability analysis. The m_stab files also use a macro. The m_stab files use the stability macros. The stability macros allow for more control over the criteria.

I_STAB Discussion

The i_stab files are meant as an introduction to the Automated Installation Tools. The full description of the automated installation tools can be found at the second link listed above. If you go there, you will see the files can be long and detailed. For this example, we will be setting up the most simple of examples for a stability analysis. For this exercise, we are really just interested in understanding a few sections of these two files. Please read the manual sections that correspond to the sections we discuss here. We will be working with the automated installation tools in several of the exercises. By reading the manual presentation in sections, you will be able to understand all of the tools much easier.

Let us start with the CIF file. At the beginning of the CIF file after the *&device* command, there are five variables (launch, transportation, loadout, upend and lift). As implied by the name of the exercise, we are really just interested in transportation stability. So, the only variable that has been set to ".true." is "transportation."

```
&set launch = .false.
&set transportation = .true.
&set loadout = .false.
&set upend = .false.
&set lift = .false.
```

The other section of i_stab.cif that we want to familiarize ourselves with is com-

mented with the word "transportation." The section has an "if statement" so that the transportation calculations are only performed if the "transportation" variable is set to ".true.," as we have done here.

This three line section is what we want to pay attention to for our transportation stability analysis:

```
inst_tran     -wind 83.19 50 100 100 -draft 6 -trim 0.53 \
              -damage 1s 2s 3s 4s 5s          \
              -no_seakeep
```

The analysis done here uses some defaults. The barge is set in a condition where the draft is 6 ft and the trim is 0.53 deg. A phantom weight is added so that the system (barge plus cargo) is at an equilibrium, then the stability analysis is performed. The only check is that the area ratio be greater than 1.4 and that the range is greater than 36 degrees. The stability analysis is done six times: one for the intact and once for every damaged compartment listed after the option "–damage." The numbers after wind specify the wind speed in knots for the different analyses (intact stability, damage stability, vortex shedding, and structural analysis). For our exercise, we will use 83.19 knots for intact stability and 50 knots for damage stability; for now that covers all of our interests for the CIF file.

For the DAT file, we are interested in two lines at the top and a section near the bottom. If you open the DAT file, the first two command lines at the top read:

```
use_mac install
&dimen -save -dimen feet kips
```

In order to use the automated installation macros, you need to have the *use_mac install* command near the top of the DAT file. And of course, you need to tell MOSES what units you will be working in. This is done with the *&dimen –save –dimen feet kips* command.

Now, go almost to the end of the file, lines 60 through the end. Here, we tell MOSES to use one of the vessels, cbrg180, from the library. Next, we set two variables "port_nod" and "stbd_nod." If you review the file box.dat, you will see that the four points are at the bottom of the piece labeled "box." The point names are selected to help position the box on the barge. The point names are the two characters after the ∗ symbol. The points that begin with the letter "p" are on the port side and points with the letter "s" are on the starboard side. The points that end with the letter "s" are at the stern while points that end with the letter "b" are at the bow. The rest of the file box.dat will not be discussed here.

The next three lines tell MOSES to place the "box" model 42 ft from the bow with the midpoint between the starboard and port points on the centerline and 6.896 feet vertically from the barge deck. Normally the word following a "–" sign is an option. This is the exception. The –PORT_NOD and –STBD_NOD options are needed.

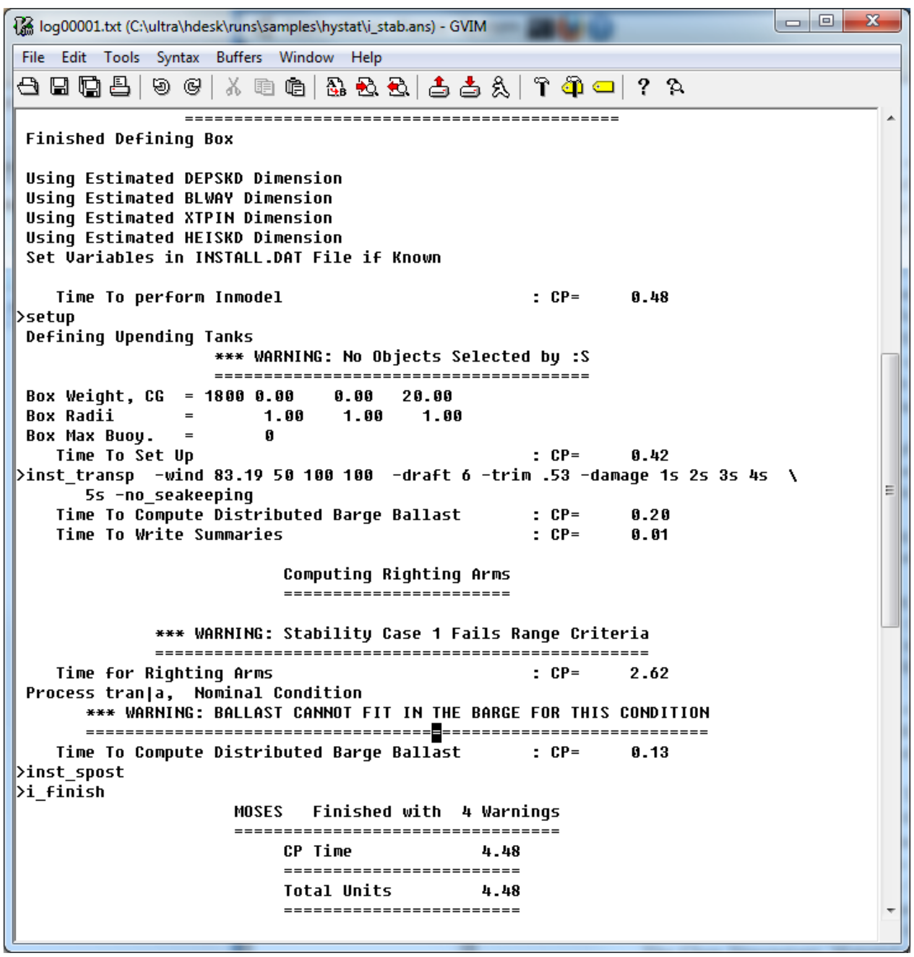
These options tell MOSES to use nodes as listed. It is important that the node order be from the stern to the bow. Or if we were to model a jacket for launch, the node order to be from leading to trailing.

The last setting tells MOSES to place the box model on top of launchways. This will cause some assumptions to be made later since we are not telling MOSES sufficient information to make a good model of the launchways.

For now, that is all we are going to discuss of the automated installation macros. Please read the sections in the manual that pertain to the sections discussed here.

We will now be reviewing the log file. In this case, the log file tells you a bit more about what is going on. It tells you that some dimensions were needed, but since they were not provided, MOSES made an estimate. It tells you the weight, center of gravity, and radii of gyration of the model "box."

The section below the *inst_tran* command tells us the results of the intact stability analysis. We know from the log file that the analysis failed the range criteria, and it lets us know that the ballast needed cannot fit in the barge.



log00001.txt (C:\ultra\hdesk\runs\samples\hystat\i_stab.ans) - GVIM

File Edit Tools Syntax Buffers Window Help

```
                    =============================================
 Finished Defining Box

 Using Estimated DEPSKD Dimension
 Using Estimated BLWAY Dimension
 Using Estimated XTPIN Dimension
 Using Estimated HEISKD Dimension
 Set Variables in INSTALL.DAT File if Known

    Time To perform Inmodel                          : CP=      0.48
>setup
 Defining Upending Tanks
                       *** WARNING: No Objects Selected by :S
                       ======================================
 Box Weight, CG  = 1800 0.00    0.00   20.00
 Box Radii       =        1.00    1.00    1.00
 Box Max Buoy.   =        0
    Time To Set Up                                   : CP=      0.42
>inst_transp  -wind 83.19 50 100 100  -draft 6 -trim .53 -damage 1s 2s 3s 4s  \
      5s -no_seakeeping
    Time To Compute Distributed Barge Ballast        : CP=      0.20
    Time To Write Summaries                          : CP=      0.01

                          Computing Righting Arms
                          =======================

                 *** WARNING: Stability Case 1 Fails Range Criteria
                 ==================================================
    Time for Righting Arms                           : CP=      2.62
 Process tran|a,  Nominal Condition
       *** WARNING: BALLAST CANNOT FIT IN THE BARGE FOR THIS CONDITION
       ===============================================================
    Time To Compute Distributed Barge Ballast        : CP=      0.13
>inst_spost
>i_finish
                       MOSES   Finished with  4 Warnings
                       ================================
                            CP Time              4.48
                            =======================
                            Total Units          4.48
                            =======================
```

Figure 18: Messages in the log file

Now, let's look at the out file. This first section (through page 10) is part of the standard output generated with the automated installation macros. The Model Size and Program Parameters let us know a little about the problem we are solving. The External Piece Summary tells us a bit about the barge CBRG180 piece MAIN and the piece BOX. We see that the barge does not attract wind loads; and therefore, our stability may be underestimating the wind arm. The Category Summary for Selected Parts tells us the same things that the log file did about the BOX model. The Class Dimensions, Material and Redesign Properties, and Class Section Data, tell us about the parts MOSES had to estimate. In the DAT file, we simply told MOSES to put in a launchway. MOSES did not have sufficient information to make an appropriate model. There were messages about this in the log file, and here are the results. For this analysis, we are doing stability and we do not care about the stress. In later exercises when we are doing a structural analysis, we will care more about these reports.

The next report, Restrain Summary, we do care about. Here is one way we can make sure that the box was placed where we asked it to be placed. We can see that *PB and *SB were placed 42 feet from the bow, each is equidistant from the centerline (the midpoint between them is on the centerline) and they are 20.89 feet from the keel. The first two are easy to check on, but what about the 20.89 feet? If we go to:

http://bentley.ultramarine.com/hdesk/tools/vessels/cbarges/cbrg180.htm

we see that the depth of the barge is 14 ft (the distance from the keel to the deck). We told MOSES to put the box 6.896 feet above the keel. So, we get 14+6.896 = 20.89 feet.

For the next several pages we get Righting Arm Results tables. The table headings let us know if it is intact or damaged. The damaged cases include a line which reads "Compartment Flooded are: XX" to indicate the damaged condition. Please notice that all of the roll columns begin with the value 0. Also, please notice that table headers include a line that reads "Initial: Roll = X.xx, Trim = X.xx deg." These initial values are the equilibrium position. The values in the Roll and Trim columns are referenced to the initial values. For the results where 1s is damaged the righting arm when the vessel is at (2–1.44=) +0.56 deg roll is 0.35 feet.

After the stability results are reports summarizing the configuration. If you recall, these are the results of the *&status* command. The following associates the command with the table header:

&status configurre	Current System Configuration
&status b_w	Buoyancy and Weight for CBRG180
&status force	Forces Acting on CBRG180
&status compart	Compartment Properties
&status category	Category Status for Selected Parts
&status draft	Draft Mark Readings

Table 1: Command to Report Association

M_STAB Discussion - Stab_ok

Let's first review m_stab.dat.

The first command *use_mac stab* has a similar format to that which we found in i_stab.dat *use_mac install*. This command tells MOSES that we will be using the stability macros. MOSES comes with a set of tools for the more frequent analyses. Among these tools are stability, finding allowable KG, finding allowable deck load, and transportation analysis. We will be working with the stability macros.

The command *use_ves* enables us to use a vessel from the vessel library.

Then, we define two nodes ∗bcg and ∗ccg. In MOSES, part of the geometry is defined with nodes, and node names begin with the ∗ symbol. Please see the following link for a more complete discussion on defining nodes:

http://bentley.ultramarine.com/hdesk/ref_man/geometry.htm#∗

Next, we define areas with their centers at *bcg and *ccg. Notice that the names given to the nodes are your choice. Here bcg stands for "barge center" and ccg stands for "cargo center." Point names can be up to 8 characters, but the "∗" counts as 1, so you choose the 7 characters after the "∗."

Please notice that we defined the area perpendicular to the x direction and the area perpendicular to the y direction in different lines. The area perpendicular to the y direction is 180*7 ft^2 for the barge and 80*75 ft^2 for the cargo. The area perpendicular to to the x direction is 50*7 ft^2 for the barge and 50*75 ft^2 for the cargo. In the following exercises, you will be asked to verify the above two statements.

Towards the end of the file, four more points are defined as non–weathertight points. That concludes the DAT file.

The commands in the CIF file should look familiar. We see two instances of the *&status* command. In the first, "B_W" is reporting the buoyancy and the weight, and in the second "nwt_down" the location of the non–weathertight points is being reported.

We already have learned that the command *&instate* sets the current condition.

The command *&weight* determines the weight so that the current condition is at equilibrium, and the *HSTATIC* puts us in the Hydrostatics Menu.

The two new commands are *stab_ok* and *kg_allow*. The command *stab_ok* asks MOSES, “Is the stability OK?” The command *kg_allow* asks MOSES, “What is the allowable KG?”

The stab_ok macro is used twice. The first time is for intact stability and the second time is for damage stability. Note that in the second time, we are damaging compartment 5p (see option –damage 5p). In both cases, we are checking that the intact area ratio be larger than 1.4 and that GM is positive.

Output Discussion

The log file has the familiar buoyancy and weight report. The height of the non–weathertight downflooding points is 13 ft. This stands to reason since they were defined at 20 ft above the keel and the current draft was 7 ft.

Notice that for the stability results of the intact case two criteria are reported, and for the damage case only one criterion is reported. This is because for the intact case two criteria were defined, and for the damage only one was defined. Please keep in mind when using this macro that MOSES will only check what it is asked. MOSES does not have an automatic list of criteria to check.

Exercise A

Use the same wcomp files as in the Basic Stability exercise.

The following are excerpts from the CFR. Each unit must be designed:

1	to have at least 1 inches of positive metacentric height.
2	Area (a) >= K * (Area (B)) K = 1.4 Area (A) is under the righting moment curve between 0 and the second intercept angle Area (B) is the area under the wind heeling moment curve to the second intercept
3	Area under the righting arm curve up to the angle of maximum righting arm equal to or greater than 15 foot-degrees

Table 2: Stability Criteria as it Appears in the CFR

The following is how they map to the options presented in the *stab_ok*:

1	The GM must be greater then IGM (or DGM)	IGM >= 0.083
2	The area under the righting moment curve will attain a ratio with the area under the wind heeling moment curve of at least IRATIO (or DRATIO), with both measured at the lesser of the downflooding angle or second intercept.	IRATIO >= 1.4
3	The area under the righting moment curve up to the angle where the righting arm is maximum is at least IARE@MARM (or DARE@MARM) ft-degree or m-degrees.	IARE@MARM >= 15

Table 3: Stability Criteria and Option Variable

Questions

Check the three criteria above for both the intact and damage stability. Check these for a draft of 7 ft.

1. Check intact stability for 100 knot wind. pass/fail ?
2. Check damage stability for 50 knot wind, damage compartment 5P. pass/fail ?

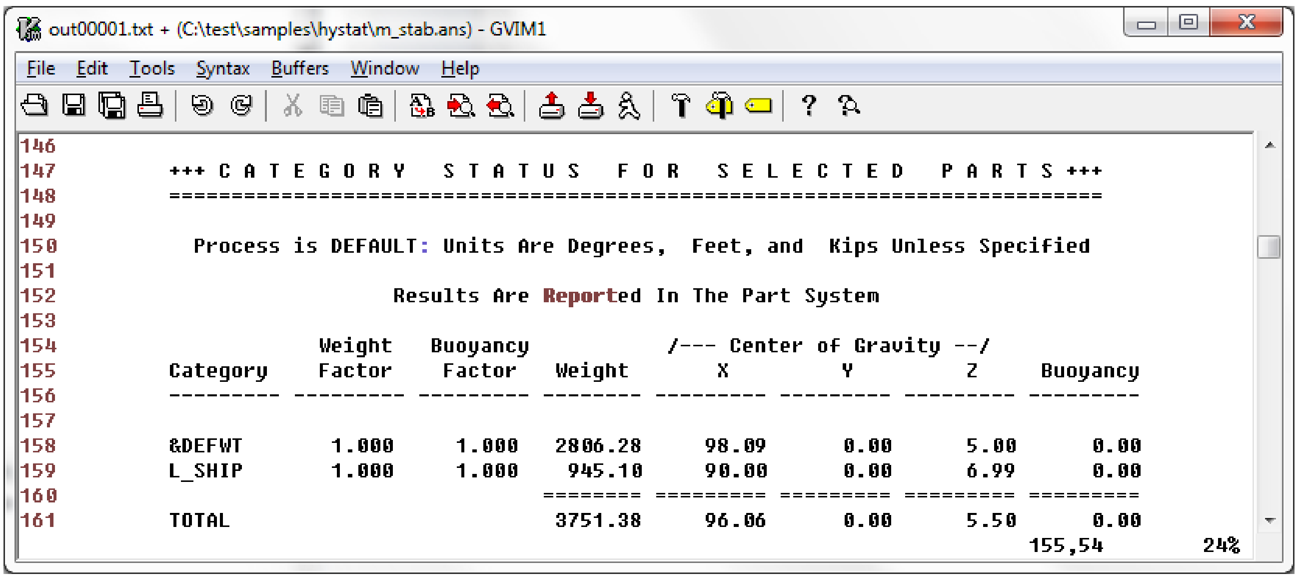
M_STAB Discussion - KG_ALLOW

The documentation of the Allowable KG command is at the same location as STAB_OK.

The last part of m_stab.cif file finds the allowable KG. We have to be careful about what center is being referred to. If you review the log file, you will see that MOSES iterated many times and reports the allowable KG to be 20.93. If you review the out file, you see that the subtitle reads VCG = 20.93 ft and the table header reads KG = 15.61. There seems to be some confusion here. What is being referenced with VCG, and what is being referenced with KG? Add the following before and after the *kg_allow* command:

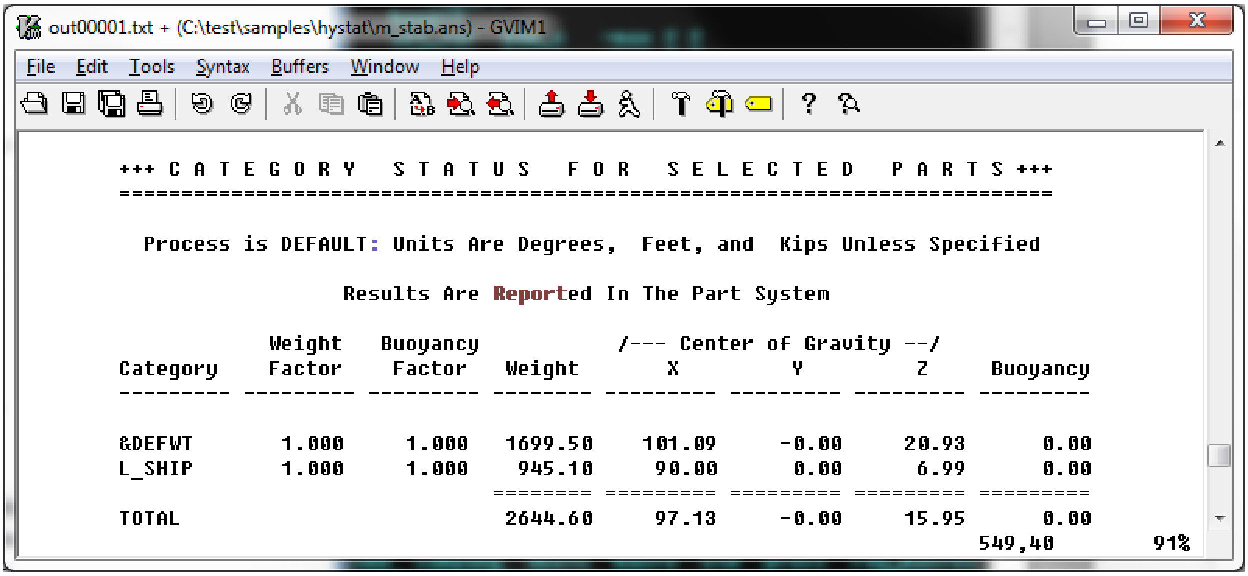
```
&status categ -both
```

Now when you re-run that analysis, there will be two more Category Summary for Selected Parts reports in the output, as shown below:




```
+++ C A T E G O R Y   S T A T U S   F O R   S E L E C T E D   P A R T S +++
===========================================================================

  Process is DEFAULT: Units Are Degrees,  Feet, and  Kips Unless Specified

                   Results Are Reported In The Part System

             Weight   Buoyancy           /--- Center of Gravity --/
Category     Factor    Factor    Weight       X         Y         Z      Buoyancy
---------  --------- --------- -------- --------- --------- --------- ---------

&DEFWT       1.000     1.000   2806.28     98.09      0.00      5.00      0.00
L_SHIP       1.000     1.000    945.10     90.00      0.00      6.99      0.00
                               ======== ========= ========= ========= =========
TOTAL                          3751.38     96.06      0.00      5.50      0.00
                                                                155,54        24%
```

Figure 19: Results Before the KG_ALLOW command




```
+++ C A T E G O R Y   S T A T U S   F O R   S E L E C T E D   P A R T S +++
===========================================================================

  Process is DEFAULT: Units Are Degrees,  Feet, and  Kips Unless Specified

                Results Are Reported In The Part System

            Weight    Buoyancy           /--- Center of Gravity --/
Category    Factor    Factor    Weight      X         Y         Z       Buoyancy
--------- --------- --------- -------- --------- --------- --------- ---------

&DEFWT       1.000     1.000   1699.50    101.09     -0.00     20.93      0.00
L_SHIP       1.000     1.000    945.10     90.00      0.00      6.99      0.00
                              ======== ========= ========= ========= =========
TOTAL                          2644.60     97.13     -0.00     15.95      0.00
                                                             549,40          91%
```

Figure 20: Results After the KG_ALLOW command

The report before *kg_allow* shows the weight &DEFWT with a ZCG of 5.00 ft, the report after the *kg_allow* shows the weight &DEFWT with a ZCG of 17.12 ft, and the VCG of the barge system at 15.95 ft. This answers the question from above. If we want to know the allowable KG for the system, we have to somehow turn off the L_SHIP weight.

Exercise B

Find the allowable KG of the system. You are looking for an allowable KG for a 6 ft draft. Check both intact and damage (comp. 5P) and for 0 and 45 degree yaw.

You will need the following command to turn off the category l_ship:

```
&apply -percent -cat l_ship @ 0
```

2.6 Longitudinal Strength

Topics:

- Parts: defining and positioning
- Difference between equi and &equi
- Define a point load and a distributed load

Reference files

p_m.cif p_m.dat

Discussion

The reference files are in the test directory under the hydrostatics tests. The reader should be familiar enough with the web page at this point to be able to locate these files and place them in the directory in which they will be working.

Both of these files are rather short. Most of the discussion will focus on options for the command we are already familiar with. We will start by discussing the DAT file. The first new option we see is the *–location* and the *–section* being used with the command *&describe body*. The reader can find the manual page that presents this command at:

http://bentley.ultramarine.com/hdesk/ref_man/bod_par.htm#&DESCRIBE BODY

Please note that we could have used the line continuation and placed both options after one *&describe body* statement. Both methods are acceptable.

Let's discuss the *–section* option, even though it appears second in the file. The *–section* option simply tabulates the section properties of the vessel. The first number is IE. Please note that the units would be the large force – length squared ($kips - ft^2$, $mtons - m^2$, or $KN - m^2$). Remember, I is in $length^4$ and E is in force per length squared. The second set of entries is the longitudinal location and the section modulus at that location. Remember, section modulus is in ft^3 or $meters^3$.

This establishes the properties of the barge. The *–location* option used prior to this simply tells MOSES the longitudinal locations. We want the results of the strength calculation reported.

The next new option is the *–desc* on the *pgen* command. Here, we are simply giving a bit of description to the part that is being generated. Remember, PGEN stands for Part Generator.

The last section may be new, but for those adventurous types that looked through the transportation macros in more depth, this may be familiar. In the last section, we describe a part that we name "cargo." As the name implies, it is just cargo, not

very descriptive. The cargo is simply a distributed weight. When we look at pictures, all we will see is the node. The weight is at x = 0, y = 0, and z=0, and the weight is distributed from –0.5 ft to +0.5 ft. The distribution was done with the *–ldist* option.

The only other thing worth noting is that the part cargo is being defined within the body barge coordinate system.

It is a good idea to run the analysis at this point. After running you will also have an answers (p_m.ans) directory. In the answers directory you will have the pictures, the log, and the output file.

Now, we are ready to discuss the CIF file. The first new command is *&describe part cargo –move 200*. We are moving the cargo from the location x = 0 to the location x = 200. Please note that we did not move it transversely nor vertically. When we view pictures, we are going to have to look at the keel to find the node belonging to cargo. In the log file you will notice that there is a message,"INFORMATION: Part Connectors May Be Incorrect". This message appears any time a part is moved via the command file. Our analysis does not include connectors between the parts. We are going to ignore the message.

We have seen the *&weight –compute* command. We are familiar with using the *hstatic* command to enter the Hydrostatics Menu. This is the first time we see the *moment* command.

Exercise A

Here we have a similar structure, as with the *RARM* command we saw in Basic Stability Exercise. The moment command does have options, but we are just using the defaults. Please consult the manual to find out more about the options. It is assumed that the user can find this manual page on the *moment* without there being a link provided here.

Let's review the new part of the output, specifically the Longitudinal Strength Results. Is the shape of the curve what we expect? Let's also examine two simple force distributions to see what is being presented.

First, let's evenly distribute the weight of the barge and the cargo weight along the entire length of the barge. Since our barge is a long rectangular shape, the buoyancy force is evenly distributed along the length of the barge.

1. Review the output of p_m. Make sure you have the shear and bending moments reported. You might even want to have a plot of this.

2. Copy p_m.cif to testa.cif.

3. Add at the top of the file *&device –auxin p_m.dat*.

4. Add *&status b_w* after the *equi_h* command, to get bouyancy and weights acting on the body.

5. Run the file and review the log file.

You should see that the *&weight* command added 20589.23 kips at x= 200 ft, y = 0 ft, and z = 30 ft.

Copy testa.cif to testb.cif. Substitute the following for the *&weight* command.

```
medit
&describe body barge
&describe part barge
*bcg 200 0 30
♯weight *bcg 20589 32 129 129
end_medit
```

1. Comment out the *&weight* command in the CIF file.
2. Run this analysis.

Compare the results in the log file. Note the results with p_m2 have a larger values. The plots do at first glance appear to be the same, but when you focus on the values you see that the loading conditions have to be different.

We are going to try several variations of the ♯weight command to see if we can figure out what the *&weight* command is doing.

Exercise B

1. Copy the p_m.dat file to testc.dat.
2. Copy the p_m.cif file to testc.cif.
3. Before the &describe part command, add the weight commands to read as follows in the dat file:

```
*bcg 200 0 30
#weight *bcg 20589 32 129 129 -ldist 0 400
```

This again changed the shear and bending moments. Here we know that the 20589 weight is being modeled as evenly distributed from bow to stern.

Do the shear and bending moment curves also show this?

Exercise C

1. Copy the p_m.dat file to testd.dat.
2. Copy the p_m.cif file to testd.cif.
3. Change the distribution of the cargo weight. See changes below:

```
*cen 0 0 0
#weight *cen 5000 -ldist -200 200
```

What this does is distribute the weight of the barge and the cargo evenly between x location -200 and +200.

Exercise D

1. Copy the testb.dat file to teste.dat.
2. Copy the testb.cif file to teste.cif.
3. Change the weight commands to read as follows:

```
*bcg 200 0 30
#weight *bcg 20589 32 129 129 -ldist 180 220
```

2 Change the cargo weight commands to read as follows:

```
*cen 0 0 0
#weight *cen 5000 -ldist -20 20
```

What is the buoyancy force per length?

Exercise E

1. Copy the testb.dat file to testf.dat.
2. Copy the testb.cif file to testf.cif.
3. Change the weight commands to read as follows:

*bcg 200 0 30
#weight *bcg 20589 32 129 129 -ldist 0 400

2 Change the cargo weight commands to read as follows:

```
*cen 0 0 0
#weight *cen 5000 -ldist -.5 .5
```

Compare these results to the original results from p_m.cif. What is the distribution from the &weight command?

2.7 Longitudinal Strength Part 2

Topics:

- Flexible barge modeling
- Longitudinal strength calculations in the structural analysis menu

Reference files

long_str.cif long_str.dat

Discussion

This set of files serve as the introduction to structural analysis. We will be comparing the results of the bending moment and shear calculations from traditional naval architecture to the results from the structural analysis method. In the structural analysis method the barge is modeled as one long beam.

The reference files are in the samples directory under the how_to directory. The reader should be familiar enough with the web page at this point to be able to locate these files and place them in the directory in which they will be working.

The top part of the dat file should be familiar. This is the section that stores the incoming units and defines the outer shell of the barge. It begins with *&describe body barge* and ends with *end pgen*. For this barge the bending moment and shear will be calculated at each station listed after *plane*. This is because the *–location* was not used. The *–section* option is needed so that the section modulous and IE can be used in calculations. For a review of the options used for a classical naval architecture bending moment and shear calculations, please see the following link.

http://bentley.ultramarine.com/hdesk/ref_man/bod_par.htm#&DESCRIBE BODY

Figure 21, shows the panel model created at the conclusion of the *pgen* command.

The next section we will describe the structural portion of the model. This begins on line 36, also with *&describe body barge*.

The command on line 37 reads *&default –nuse* @. The @ is the wild character in MOSES. This is the first time we see this command. The option is *–nuse* and the other common one is *–use*. These are used to turn on, *–use*, or "not use" turn off, *–nuse*, the attributes. The attributes are weight #dead, wind #wind, added mass #amass, drag #drag, and buoyancy #buoy. Using the @ sign means we are turning

them all off. The reference for the attributes is at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/bod_par.htm#&DESCRIBE BODY

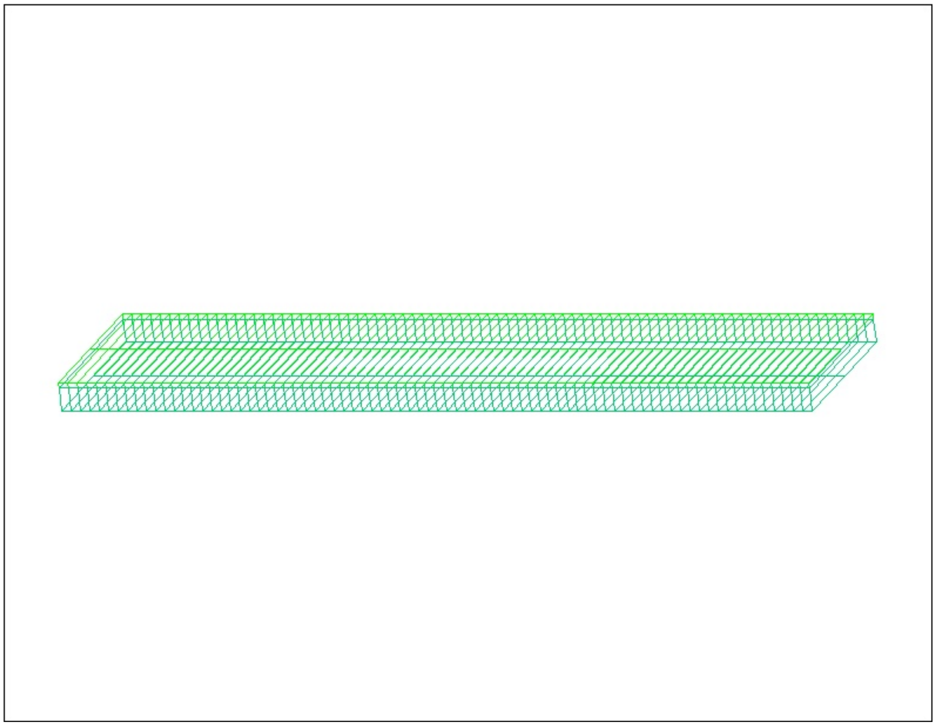


Figure 21: Iso view of panel model

Here we will be defining a class with the ˜ character. Similar to the @ character, described earlier, the ˜ is a special character in MOSES. The ˜ denotes class names. When we define the class we will also be defining its properties. We are interested in the command on line 48,

```
~bardum pri (21000*2) 15000 -section 4.398e8 5.65e16 7.854e16 9.6304e16
```

The command can be found at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/bod_par.htm/cls_str

In the page for class structures we can find the general dimension of a PRI class. The values we have used here for the PRI are the gross dimensions of the barge. We need to make sure that the same section properties are used for the classical naval architecture method and for the structural analysis method. In the structural analysis method we are interested in the IY term because this is used for loading along the barge centerline in the global z–direction.

If we review the documentation for the *&describe body* command we see that EI is in terms of bforce-m^2 units. Since we are in meters and kNts this would be kNts-m^2. If we review the element properties documentation we see that for a class definition *–section A IY IZ J* α_y α_z use mm^4 for IY and IZ. What we know is that

$$IE = 113 * 10^11 kNts - m^2 \qquad (2.7 \qquad 1)$$

and

$$E = 200 * 10^6 \frac{kNts}{m^2} \qquad (2.7 \qquad 2)$$

therefore,

$$I = \frac{113 * 10^{11}}{200 * 10^6} \frac{kNts - m^2 m^2}{kNts} = 56,500 m^4 \qquad (2.7 \qquad 3)$$

Which then gives us $I = 5.65 * 10^{16} mm^4$ This is the value used for IY in the *–section* option. This completes the section properties definition.

The rest of the structural model is a beam along the centerline. Our intention is to compare the results to the results we will calculate with the traditional naval architecture method. We therefore need to make a structural model with at least the same number of elements as the number of stations used in the *plane* command. The structural model is shown here as a loop.

The loop begins with the command *&loop* and ends with the command *&endloop*. Before the loop we have to initialize the "list" variable. The two other variables "num_beams" and "dist" need to multiply out to the length of the barge. If there are going to be 80 beams then we need to make them 5 meters in length. Meaning we cannot change "num_beams" without changing "dist".

Now back to the loop. Basically the loop makes the nodes then places a beam element between each set of two nodes. Towards the bottom of the loop there is a section where the weight per foot ($16070 \frac{kNts}{m}$) of the element is defined. Let's examine each command.

The opening of the loop command reads

```
&loop lll 1 &number(real %(num_beams)+1) 1 .
```

The format of the command is

```
&loop INDEX BEGVAL ENDVAL INCREMENT.
```

In this case the "lll" is the index, 1 is the beginning value, 81 is the ending value, and the increment is 1. Notice that the math $80 + 1$ is done as part of the reading. We need 81 elements because there are 80 elements, that means that the last element connects node 80 to 81. The first variable is "kkk" for the beam number. The beams are going to be labeled by number from 1 to 80. Notice that the first time through the loop kkk = 0. But a beam does not get created until lll is greater than 1. This is done via an if statment *&if %lll .gt. 1. &then*. So the first time through the loop

the only thing that really gets done is

kkk = 0
xl = 0
name = n00
last = *n00

in short, a node *%(name), "*n00", is created. The second time through the loop

kkk = 1
xl = 5
name = n01
beam n01 ~bardum *n00 *n01
last = *n01

This similar pattern is repeated 10 more times until lll is greater than 11. When lll = 12

kkk = 11
xl = 55
name = n11
beam n11 ~bardum *n10 *n11
#elat n11 16070 0 0 0 0 -cat elat
last = *n11

This similar pattern is repeated while lll .lt. 72. Once lll = 72 then the #elat definition is taken out.

Once lll reaches 80 the loop finishes. The last command remembers the units that were in place at the start of the file. Figure 22 shows the structural model of the barge.

In the structural model each element was defined with an element attribute command *#elat*. What we are defining here is the weight per foot of each element. This complements the *&default –nuse* @ command we mentioned earlier. Because the barge will get the buoyancy force from the panel model generated with the definition using *pgen*, we do not want the structural elements to also contribute to buoyancy.

The manual page for the command *#elat* can be found at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/ele_beams.htm#ELAT

Figure 22: Structural model

This concludes the model for this exercise. We have defined a model to be used with the traditional naval architecture approach and the structural analysis approach.

Command File Discussion

Now we will discuss the command file. The first two commands *&device –g_default file* and *&dimen –dimen meter kNts* you will see very frequently. With *&device –g_default file* we tell MOSES to send any plots or views of the system to a file in the ans directory. With the *&dimen -dimen meter kNts* we set the dimensions that are going to be used for the analysis.

In the next commands the barge is put at a 12 m draft, 0 trim, and 0 heel. This part is similar to what was done in the p_m files.

The next section is model editing. This begins with *medit* and ends with *end*. We are going to add restraints for the structural analysis. In the data file we have added the weight. The weight is a force in the negative z direction. When the vessel is placed in the water there will be buoyancy forces in the positive z direction. For the structural solution to solve, we need to restrain the ends.

Just like in the data file we first define the class. Here we are defining a linear spring

with ˜spr1 and ˜spr2. We are defining the stiffness in the x, y and z directions for ˜spr1. We are defining the stiffness in x, y, z and rx for ˜spr2. For the structural solution just one restraint in roll is needed for a solution to be found. The restraints are connected to the first node and the last node. In both cases the nodes are connected to the barge structural model at one end and to ground at the other end.

This is all the model editing we have to do. It is a good idea to wait and model the restraints in the command file. This way the model can be placed at the desired orientation, in our case the proper draft, roll and trim, before any connectors to ground are added.

A picture is generated with the *&picture iso* command. There really is no good or bad reason to place this command here. It is always a good idea to provide a picture of your system. The author of the command file simply chose this as the place.

In the next set of commands we review the status up to this point, then we alter the mass properties such that the system will be at equilibrium for this condition. This is in the lines that are set off with the comment "compute weight". In this section we are going to also be discussing the values in the log file. It would be a good idea to run the analysis and have the log and out file available in the answers (ans) directory.

The first command *&status config* reports the status of the configuration. It summarizes the location and the net force. The results show that the barge is at 12 m draft. The forces are reported in the local body system. For this barge the origin of the body system is at the intersection of the bow, centerline, and keel. The forces show that there is a net positive force in the z direction and a net negative moment about the y axis. This stands to reason. We have only defined a weight for the structural beam model of the barge. The positive z direction force tells us that the buoyancy force is larger than the weight force. The buoyancy force is probably located near amidships. This would create a negative moment about the body y axis.

When we review the results of the command *&status force* our above reasoning is validated. We see that the inertia force is -6763 mtons. This means that in order for the sum of the forces in the z direction to equal zero, a force equal to -6763 mtons would need to be added, or a weight of 6763 mtons would need to be added. Please be careful in interpreting the values in the "Inertia" row. These values represent the inertia needed to put the system in equilibrium. When the system is very near equilibrium the values in the "Inertia" and "Total" rows will be small.

Up to this point we have established that we are not in equilibrium. The command *&weight -compute bname zcg kx ky kz* will get us in equilibrium. A word of caution is warranted here. MOSES will place a weight on the xy plane where z = zcg. This weight will make the $\Sigma F = 0$ for all three degrees of freedom and $\Sigma M = 0$ for all three degrees of freedom. There is no checking to make sure the location of the mass is within the defined barge body. This command was also part of the p_m files

discussed as the first part of this exercise.

Now we check that the system is indeed in equilibrium with the command *&equi* is used. The results of *&equi* show that the net forces and moments are 0. This is also confirmed with the results of the *&status force* command. Here the results show the inertia and the total rows having values near zero.

The next set of commands are there to further satisfy our engineering need to show the details of the analysis. Also, for many projects the input values are part of the report. Knowing how to get MOSES to report the input values as part of the output will help in validating the results. *&status b_w* reports the status of the buoyancy and weight of the body. *&status cat* is short for *&status category*. Recall that in the dat file the weight distribution we assigned to each beam structural element was labeled as "elat". When we look at the "Category Status For Part Barge" report we see the structural beams weigh 4.821E6 mtons labeled as ELAT and the weight added 6762 mtons labeled as &DEFWT. When the *&weight –compute* command is used the default is to label the category as &DEFWT.

In the next section we also report more details on the model via the summary menu. This menu is entered with the command *&summary* and ends with the command *end*. This reporting goes to the output file. The first set of reports deal with class properties. This creates the "Class Dimensions Summary", "Material and Redesign Summary", and the "Class Section Summary". These are pages 3 to 5 of the output. When you review them you will see the values we used in the data file to define the structural model.

The second command in the summary menu is *compart_sum long*. This creates the Vessel Section Properties for Barge report. When you review the report you will see values we used in the *&describe body barge –section* command in the data file.

In the next section labeled Traditional Long. Str. the same approach as was presented in the p_m files is taken. The Hydrostatics Menu is entered with the *hstatic barge* command, and is exited with the *end* command. The bending moment and shear is computed with the *moment* command. This puts us in the Disposition Menu. In the p_m files all we did was make the standard report with the command *report*. Here we also generate, see Figure 23 with the command *plot 1 2 -rax -t_sub "Traditional Results"*. In this set of files we are interested in making a comparison. So we have to somehow store the values for later use. This is what is done in the lines beginning with the command *set*. Here a variable is being defined but is not exactly like the command *&set*. The command *set* is only valid in the disposition menu and serves to store data that is currently available for reporting. The following is the link to the

manual page on the *set* command.

http://bentley.ultramarine.com/hdesk/ref_man/disp_set.htm#SET_VARIABLE

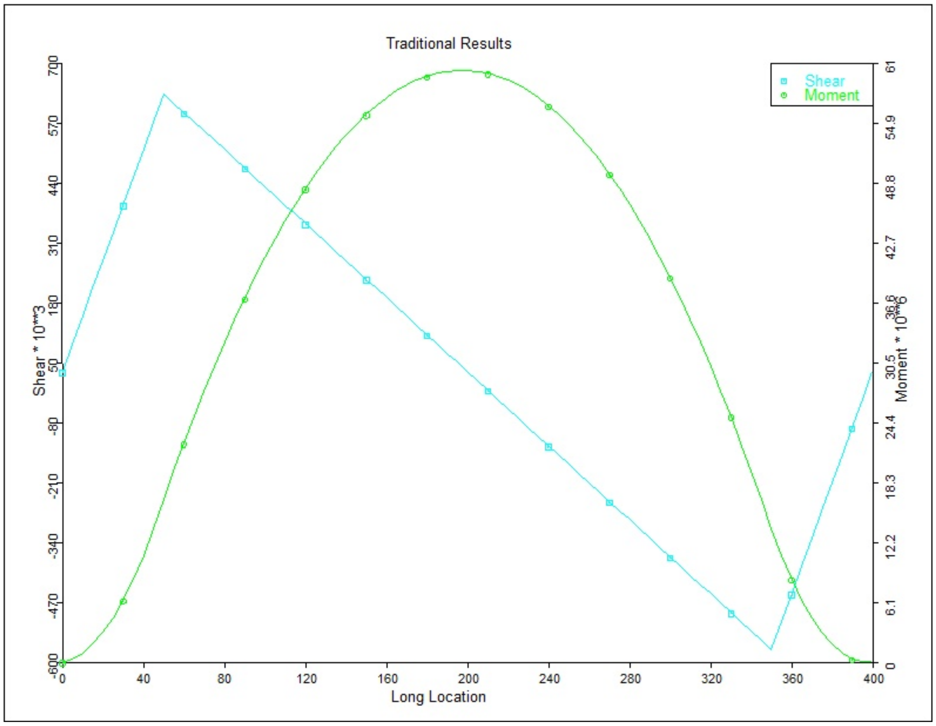


Figure 23: Traditional Shear and Bending Moment Results

The comments in the file tell us what is being stored. In variable "shr" we are capturing location and shear values. In variable "bmom" we are capturing location and moment values. In varible "defl" we are capturing location and deflection values. In the last comment *store* the data from columns 1(location), 2 (shear), 3 (bending moment), and 5 (deflections) will be saved in the answers directory as a comma separated file (file ending in csv).

The first *end* takes us out of the disposition menu. The second *end* takes us out of the hydrostatics menu. This concludes the traditional naval architecture part of the analysis.

The structural analysis begins with the comment line "structural analysis". The first command is necessary, although not obvious. We have to declare a name for the current environment. This is a still water analysis meaning there are no environmental forces. The command *&env none* makes an environment labeled "none" with zero environmental forces. With an environment declared we can now perform a structural analysis.

The structural analysis menu is entered with the command *struct* and is exited with the command *end*. Inside the structural analysis menu we first create a load case based on the current status condition with the environment associated with "none". This is done with the command *lcase –static none*. The structural solution is solved

for with the command *ssolve*. Inside the structural analysis menu the solved solution is calculated. We have to go into the structural post–processing menu to report the values or to further manipulate them.

As part of the structural post processing we will be making the comparison of the traditional method to the structural analysis method. Before we start structural post processing we need to further process the data we collected earlier. First there is sign change to be made. In traditional naval architecture bending moment is positive for a sag case. In structural analysis the bending moment is negative for a sag case. The two sections with loops, set off by the the comment "change sign of traditional results to match coordinate system of structural analysis" is where the signs are changed.

Before the loops are started, an initialization of the variable "n_shr" is performed. *&set n_shr* simply make the variable "n_shr" exist. This variable is used in the loops. When the loops are first started n_shr is empty. If it does not exist before it is used, it will cause errors.

Remember for these variables we kept two columns. For the variable "shr" we kept the column for location and the column for the shear value. If you review the output report titled "Longitudinal Strength Results" we have:

shr = 0 29979 5 90241 10 150503 . . . 400 29981

The loop that is being used here looks at the values in multiples of 2. Let's look at

this loop similar to how we looked at the loop in the data file.

The first time through

ppp = 1
chg = 29979
chg = -29979
xval = 0
n_shr = 0 -29979

The second time through

ppp = 2
chg = 90241
chg = -90241
xval = 5
n_shr = 0 -29979 5 -90241

The final time through

ppp = 81
chg = 29981
chg = -29981
xval = 400
n_shr = 0 -29979 5 -90241 . . . 400 -29981

A similar pattern is followed for the variable bmom creating the variable n_bmom. At the conclusion we have changed the sign for shear and bending moment results from the traditional naval architecture approach. Now we need to report the values from the structural analysis method and compare them to the traditional naval architecture method.

To enter the structural post processing menu the command *strpost* is used. To exit the structural post processing menu the command *end* is used. The first report generated is the deflections. The command *joint disp –local yes –file yes* generates the "Joint Displacement" report in the output file, and the top portion of the ppo00001.txt file in the answers directory. The command *rest –detail* generates the "Restraint Load Detail", and the "Restraint Envelope Standard" reports in the output file. The command *beam load –file yes* generates the "Beam Load Standard" report in the output file and the bottom portion of the ppo00001.txt file in the answers directory. The shear and bending moment values in this report are the ones we want to compare to, however, they are not available in pre-formated tabular form. We need to issue the *bmom_shr* command to have them available in the disposition menu so that we can save the values for later comparison.

After the *bmom_shr* command we are in the disposition menu. Earlier when we were in the disposition menu we saved the value of interest into variables. Now we are going to add these values to the ones already there. We add these with the command

add_column –input CS X(1) Y(1), X(2) Y(2) . . . . When we add our two columns we will input 1 for CS because the first column in this table is distance. Which is the same concept as location that we saved. I know the first column is distance because of the results of the command *vlist* a few lines down, in the command file. In the log file, you can view the column labels at the end of the column addition, more than a few lines down.

From the results of *vlist* we see we want to compare column 4 to column 8 and column 6 to colum 9. The comparison plot for shear is shown in Figure 23, and the comparison plot for bending moment is shown in Figure 24. The plots are also in the answers directory. You will notice that for the shear comparison the structural analysis line is jagged. This was expected. In the command file there is a message: "REMEMBER: the load is distributed but the buoyancy is lumped at the nodes". The moment comparison is smoother but still off by the fact the buoyancy is at the nodes.

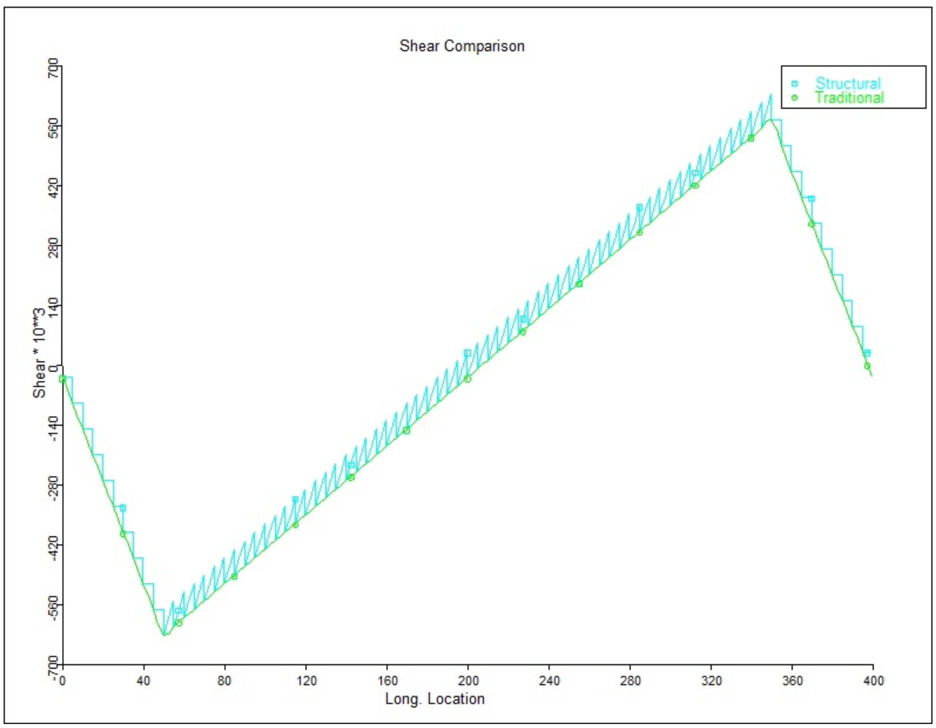


Figure 23: Comparison Shear

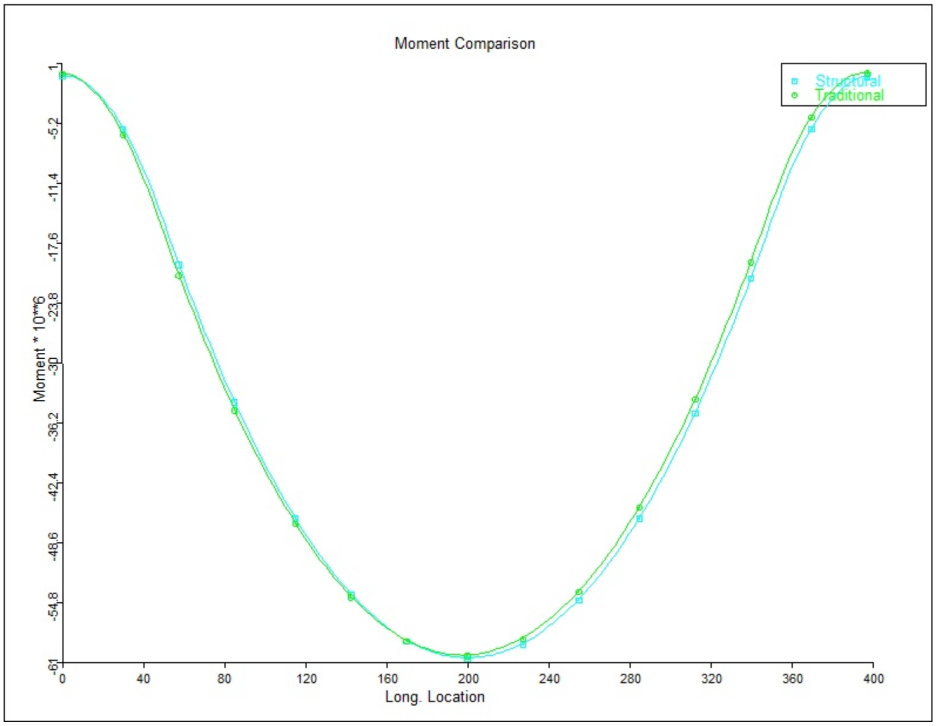


Figure 24: Comparison Bending Moment

Exercise A

Earlier we said that it is important that the panel model be the same length as the structural model. It would be nice for MOSES to report check this and report it as an error if it did not match.

Add some checks so that when the length of the structural model does not equal the panel model an ERROR is reported.

In the data file add after the panel model:

```
&set vlength = 400
```

In the data file add after the structural model:

```
&set str_len = &number(real %num_beams*%dist)
&if .not. %str_len .eq. %vlength &then
   &error ERROR Structural model length different from panel model
&endif
```

Exercise B

This exercise shows how to use some of the graphics options.

1. Restart the analysis.
2. From the main menu click on “Graphics”.
3. From the sub menu click on “Picture Options”.

4. This will bring up the Picture Options pop-up window.
5. On the "View" tab under Picture type select "Structural" and under Render Type select "Solid".
6. On the "Misc" tab make sure the box next to "Show Water and World" is blank.
7. Then, first bring down the menu with "Show Deflected Shape:" and select "yes".
8. This will add a cell for "Deflection Multiplies". Input 100.

This brings up a structural picture of the barge with the deflection magnified 100 times.

3 Convert from SACS

3.1 Translating from SACS

Topics:

- Introduction to translating from SACS
- Checking the model

Reference files

sac_tpg.cif sac_tpg.dat

ck_sac.cif ck_sac.dat

cnv_ck.cif jacket.sac

Discussion – the long way

The reference files are in the tests directory under the directory convert. The reader should be familiar enough with the ultra directory (MOSES installation directory) at this point to be able to locate these files and place them in the directory they will be using.

Here, we present the procedure to convert a SACS model to MOSES format and a suggested method of checking that the conversion was done correctly.

Converting a model is a simple exercise. It is the checking that needs to be done carefully. If you look at the sac_tpg.dat file, you will notice that the first line is a MOSES command and until near the end, all of the commands are SACS commands. The MOSES command at the top reads:

```
&convert sacs -jright 000 -cright 000
```

Please see the manual page on *&convert* for a more detailed explanation of the command and the options. The last two lines of the file are END and *&finish*. The END command is the last SACS command and the *&finish* command is the MOSES command. These two lines are all you need in the data file to convert.

Now, let's look at the CIF file.

The CIF file has three lines. The first line is really not needed for the conversion to work, it is put there to make the output easier to read. The command *&device –oecho no* tells MOSES not to echo the data file to the output file. Please recall that in most of the previous exercises, the output contained an echo of all that was read in with *inmodel*. This top section in the output can get rather large. In general,

–*oecho* should be set to *no*. The only time I set it to *yes* is if I am debugging.

You should be familiar with the last two lines.

These three lines are all that is needed to convert the SACS model to MOSES format. When you run MOSES, the new MOSES model will be in sac_tpg.ans/mod00001.txt. (This is if you did not change your defaults). This completes the converting part of the process.

In order to check that the conversion was done properly, we need to look at the ck_sac.cif and ck_sac.dat file. The check files are the minimal checks. Depending on the complexity of the model or what its intended use is, other checks may be necessary. This exercise presents the minimal checks.

If you review the DAT file, you will see that the first line is needed so that MOSES knows it is a body. These checks are done with the intent of using the models in the installation macros. Since the installation macros add the *&describe xxx* line to the model, I do not take the time here to add this line to the mod00001.txt file. In general, I do not change the mod00001.txt file.

If you need to convert a file, it would be a good idea to copy these four files to your working directory and then just change the SACS part.

Discussion – the short way

Now, take a look at cnv_ck.cif. This file is used to convert the SACS model found in jacket.sac. This uses one of our macros and it makes all of the changes and creates the check files for you. Run this CIF file and answer the following questions:

Exercise A

1. How much does LOADLC3 weigh?
2. Where is the center of gravity of the elements?
3. What is the total buoyancy?

Exercise B

Start with the sss.inp file in the samples/data directory. Convert the model and call it l_jack.dat. You will need to add the command *categ -brief* to the section within the *&summary*.

1. Where is the center of gravity of the elements?
2. What is the total buoyancy?

4 Motion

4.1 Dynamic Flooding

Topics:

- Dynamic flooding of a compartment
- Setting the initial pressure in a compartment

Reference files:

- /ultra/hdesk/runs/tests/compart/sink.cif and sink.dat
- /ultra/hdesk/runs/samples/how_to/up_damage.cif and up_damage.dat

The two sets of files are intended to complement one another. In the sink analysis, a barge compartment is dynamically flooded. In the up_damage analysis, a tubular jacket leg compartment is dynamically flooded. In both cases, the intent is to show how to model water entering a compartment in the time domain. The potential uses for this set of commands are the dynamic flooding during an upend and the accidental damage to a compartment.

Sink Dat File Discussion

Let us begin with discussing the DAT file. We have two new parts of the barge defined here: the draft marks and the valves.

The draft mark definition starts with the comment "*** draft marks." In real life, vessels have vertical lines painted with numbers so that the draft can be easily read. We are going to make several lines perpendicular to the *xy* plane. MOSES will use these lines to report the draft at that location of the vessel.

In the file, we have put a total of four draft marks. Two draft marks at the bow and two at the stern, two on the port side and two on the starboard side. The first step is to make the nodes. We have employed a naming convention, the first character is "b" for bow, "s" for stern. The second character is "s" for starboard, "p" for port, and the third character "b" for bottom and "t" for top. Making a total of eight points for four lines. Following this convention, the point *bsb is a point located on the bow starboard bottom.

To actually define the draft mark lines, we use the *&describe body* command again. Please keep in mind that you need to <u>first</u> define the body, then define the nodes to use for the draft marks, then use the *&describe body* again to further add to the body description. Here we use the *–dmark* option. Please notice that each draft mark requires its own *–dmark*.

For the *dmark* option, we need to name a bottom node and a top node. Again, we have employed a simple naming convention. The first character is "b" for bow or "s"

for stern, and the second character is "s" for starboard and "p" for port. Please keep in mind that the draft mark line must be defined from bottom to top. MOSES will measure from the first node toward the second node. **If you reverse the order, then you will be getting free board, not draft**.

The second new item we have added to the vessel description is the flood and vent valves. Flood and vent valves are holes on the outer shell that allow water to enter or exit an assigned compartment. This section starts with the comment "*** compartment." In real life, they could be vents.

Here we define a node for the vent valve and label it *vent, and we define a node for the flood valve and label it *flood. The defining of the nodes is, as done in earlier exercises, with the * symbol. By using the command *&describe hole*, we are telling MOSES that these are holes on the outer shell. For describing a hole, we tell MOSES its type, size, location, and a friction factor. If we were simply doing a static analysis, we do not need to tell MOSES the diameter nor the friction factor. In this exercise we will be doing a dynamic flood; we want the flow rate to be correct. So, we do need to make sure the areas and the friction factors are correct.

Finally, the last step in defining the valves is to tell MOSES which compartment is assigned to these holes. This is done with the *–holes* option on the *&describe compartment* command.

Sink Commands Discussion

For the exercises up to this point, we have been discussing the command file before reviewing the log and output files. For this exercise, it will be much easier if we discuss the log file and its contents as we are discussing the commands as they occur in the command file. Please run the analysis and create the log file so that the rest of this discussion can make sense.

Do not delete the database or the answers directory, but instead restart. After this restart in the sink.ans directory, there will now be a log00001.txt, out00001.txt, and a log00002.txt.

Once the MOSES window appears, type CTRL–G.

This should bring up a rendered figure of the final event. With the mouse you can move the green bar near the bottom to left near 0.0. Then, you can press the "Play" button and watch the simulation.

Now we can discuss the commands. For this exercise, we are not going to merely discuss the commands, but we are going to review them in the log file as well.

We are familiar with the commands at the top of the CIF file from previous exercises. The first new command we encounter is *&parameter*. The command *&parameter* sets many of the basic parameters in MOSES. Please see the manual page for all of the

available options.

This is the first time we are going to work explicitly with a Process. In MOSES, a process has events. We will be doing a time domain process. We need to tell MOSES how long to make the process and how many events, or how big the steps are going to be. This is what is done with the new *&env* command.

We need to define an environment because the flooding will be a time domain process. Usually when you define an environment, the description will include wind, waves, and current. We are only describing the total time and the time step, and we are naming the environment "null." Please see the manual for all of the possible options.

The next command *&compartment* we have seen before, but we have not seen the option *–dynam*. Please notice that in this command we are telling MOSES three things:

Option	Description
-correct	to calculate the CGs and the derivatives at each event in the simulation
-percent	to empty the compartment to 0 percent
-dynam	the compartment will be filled dynamically

There are many ways to model the ballast water in a compartment. The type "correct" will be used for most analyses. If you are interested in the other options, please review them under the *&compartment* command. The other methods of modeling the ballast water are so rarely used that they will not be discussed here.

A word of caution is needed here. The placement of the commands *&compartment* with the *–percent* and with the *–dynam* options should be placed immediately before the *tdom* command. If there are commands between the *&compart* and *tdom*, you run the risk of water being added to the compartments unintentionally. The *tdom* command is the time domain calculations. **For dynamic ballasting, the starting ballast and the flag to set the compartment to dynamic flooding have to be placed right before the *tdom* command**.

The actual time domain calculations are performed with the *tdom* command. Please note that the *tdom* command is a Main Menu command and at the conclusion of the time domain, you are still in the Main Menu. This is different from the frequency domain calculations where it is all performed within the Frequency Domain Menu.

As mentioned earlier, a time domain analysis is a process. In order to view the results, we need to be in the Process Post Process Menu, *prcpost*. For any analysis in which there is a process, you will be able to use this menu.

In general, the sub–menu *trajectory* should be available for all of the analyses in which there is a process. The other menus we will use here are generally only available for

similar analyses, as we have here.

The sub–menu *draft* works with the draft marks we described in the DAT file. For future analysis in which you are interested in how the draft changes with events, you will need to describe the draft marks similar to how we did here.

Since this is our first venture into this menu I will use specific examples, but in general a process can have many different configurations. The specifics that work in this example will probably have to be changed if you look at a different process.

The sub–menu categories are *tank_bal*, *hole_flood*, *tank_fld* and are generally available when you change the ballast arrangement during a process.

All of these commands get us to the disposition menu. There are standard reports for all of the sub–menu categories. Here we ask for the standard report of all of them with the command *report*. We ask for plots of the trajectories and the tank ballasting with the command *plot*. Remember we can use the *vlist* command to determine what numbers to use with the plot command. These two plots along with some bow views of the events follows.

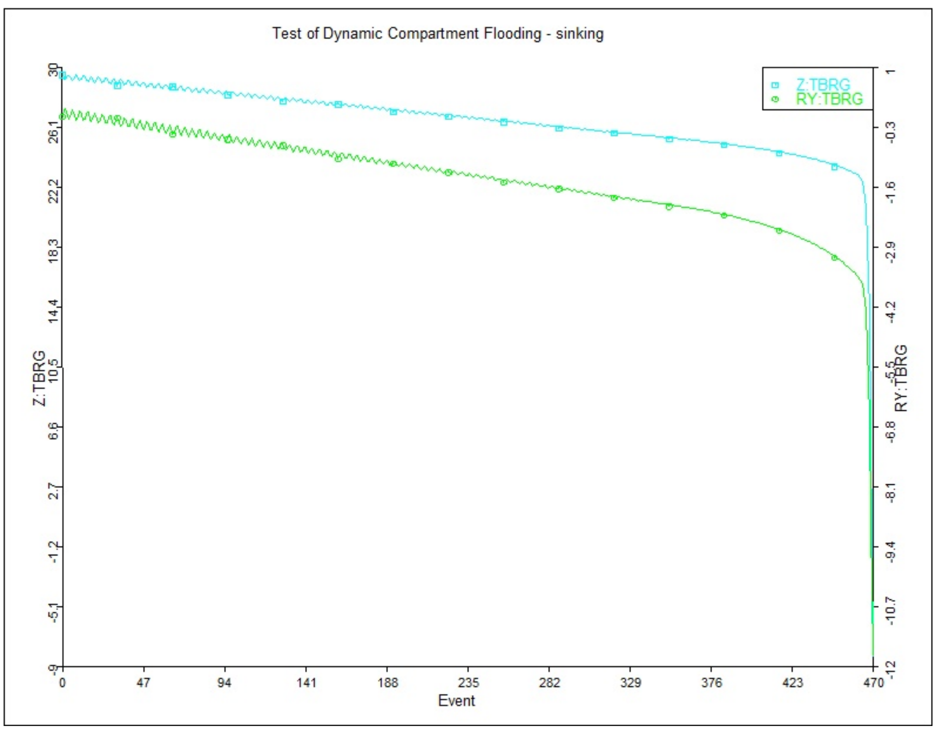


Figure 25: Z and RY motion of the Body Origin

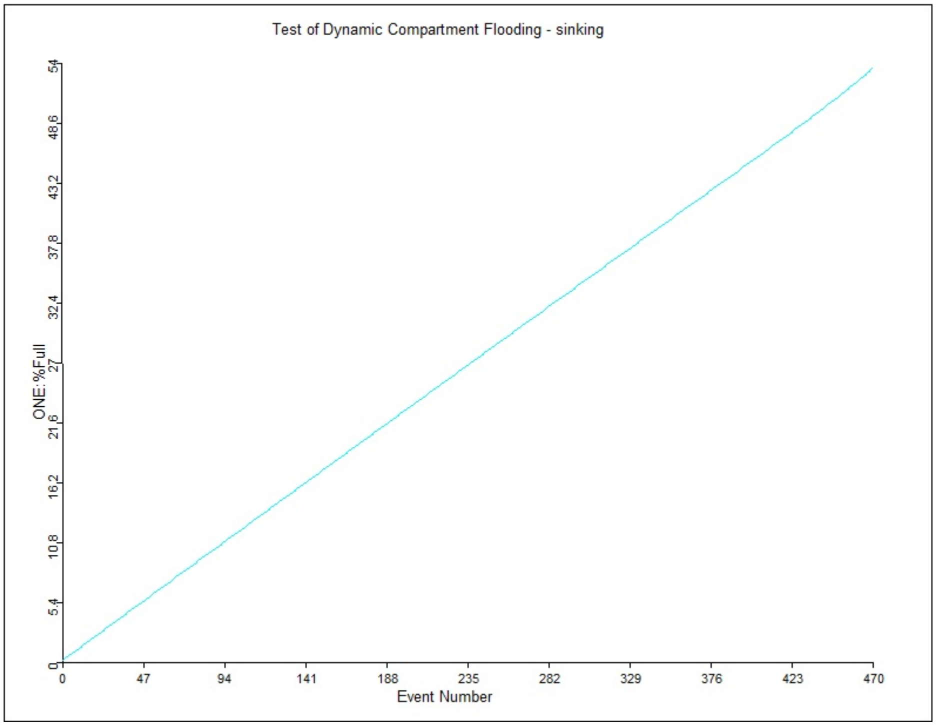


Figure 26: Compartment flooding in percentage

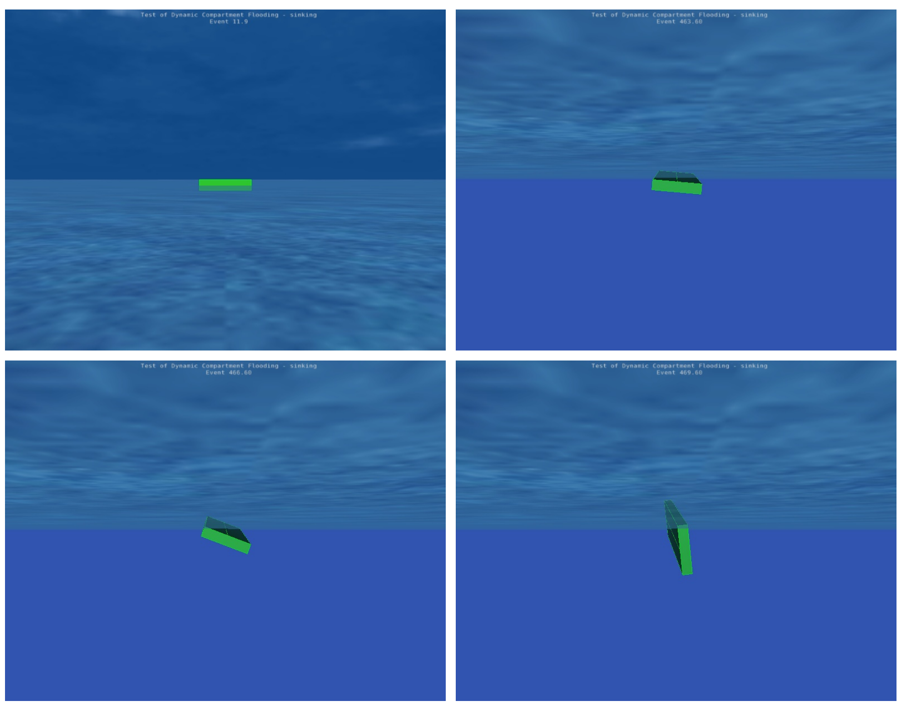


Figure 27: Events during flooding

Up_damage Dat File Discussion

The file can be divided into the four sections:

- definition of the classes, all begin with the $\sim$ character
- definition of the beams, all begin with the word *BEAM*
- definition of the joints, all begin with the $*$ character
- definition of the compartment, section at the bottom with many occurences of the *&describe* command.

The first three sections are needed to describe a jacket. The format is similar to that which one would get after translating a model from SACS format. In general, a jacket model is used when structural detail is needed. Structural detail is a very broad term and can include anything from weight distribution to joint can definition. For our analysis, it is sufficient that the jacket model has structural members that have weight and buoyancy attributes.

When MOSES reads a data file, it takes it awhile to process all of the information. Notice that the *BEAM* definition which uses the joint definition is before the joint $*$ commands. It is acceptable for the definitions to be out of order in the data file. MOSES takes the information, sorts it, then puts it together.

The compartment definitions are found towards the bottom of the file. The format is very similar to what we found in sink.dat. This is the first time we are going to use the *tubtank* command to describe a compartment. We are using the command *tubtank* to model the compartment defined by the interior of the jacket legs. The jacket legs can be described as long cylinders. The command *tubtank* is an easy and computationally efficient way to describe compartments that are cylindrically shaped.

Notice that the compartments are described in SI units whereas the rest of the jacket is described in feet and kips. The use of the *&dimen* command with the options *-save* and *-remember* make the use of mixed units easy to handle.

Up_damage Command File Discussion

Reading the model with the *inmodel* command gets us to the model edition section, *medit*. In this section, we are redefining the part coordinate system. When we review the results, it would make our life easier if roll was reported along the long axis of the jacket, and pitch was reported along the axis at the base of the jacket and perpendicular to the long axis of the jacket. Up to this point, we have not taken any effort to determine the location of the jacket coordinate system. Instead, we

are going to use the *&describe part* command to define the part system to fit our analysis.

The way we are going to use the *&describe part* command needs a bit of discussion. The command implies that you are changing the part coordinate system. We need to be clear about what the rules are with the naming convention. With a part system whose name is different from the body, then only the part system is redefined. With a part system whose name is the same as the body, then the body and the part system are being redefined.

When you read the manual page on the *&describe part* command, you will see that the order of the nodes is important. We want the x-axis to be from the bottom of the jacket towards the top so that roll is measured along the long axis. This means points 4 and 2 have to be at the base of the jacket and points 1 and 3 at the top of the jacket. We next want y to be transverse on the base of the jacket, preferably on the face floating at the water surface in the beginning event. If you look at the following picture, you see the point selection *J4003 *J4001 *J8003 *J8001 will result in the desired coordinate system. Figure 3 shows the locations of the points.

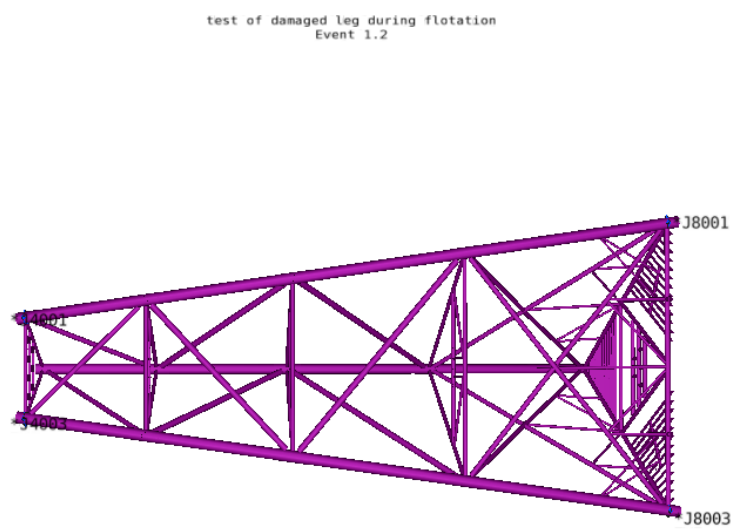


Figure 28: Points used to redefine the part coordinate system

The commands used to create the picture are:

```
&select :ppp -select *j4003 *j4001 *j8001 *j8003
&picture top -render gl -water no -points :ppp -anotate points
```

The only thing you will need to do with the mouse is set the event to 1 with the

slider bar and use the up and down arrow keys to zoom in or out as desired.

This is all we need to do as far as editing the model. In the next set of commands, we get a summary report of the compartments defined in the data file. We have seen the *&summary* menu earlier when we discussed the allowable KG macros. Here we are asking for a compartment report instead of a category report. As you can see, the *&summary* menu is a prepared way of presenting summaries of the database. Following that, we use the *&status* command to get the status of the compartments. We have seen the *&status* many times in the other exercises.

In the next set of commands we find the initial floating position. This is the first time we see the *-guess* option used on the *&instate* command. The pair of commands, *&instate* and *&equi* are needed to find the equilibrium quickly. This is similar to the previous exercises where we first set the body at a condition (draft, roll, trim), then found equilibrium. The *&instate* command orients the body and the *&equi* finds the position where the sum of the forces and the moments are very near zero. Please keep in mind that for all analyses, the program will find equilibrium in a small number of iterations the closer it is to its equilibrium position. That is to say, we could have also issued multiple *&equi* commands until the program found an equilibrium position. This plan of action would eventually get us to the same position, however we would probably be frustrated by then.

The results of the *&equi* command show that the jacket is in equilibrium with a slight roll and slight trim. That is to say the body coordinate system is oriented at a 0.03 degree roll and a -0.89 degree pitch with respect to the global coordinate system.

In the next section, we report the position of the four points we used to define the coordinate system. We are using the string function *&points*. String functions were introduced in the Working with Compartments exercise and will not be further discussed here.

The only point in reporting the positions of the points is to assure ourselves that an equilibrium position close to the one that we anticipated was found. In this section, we are reporting the position in the body followed by the position in the global. The reported positions show that there is a slight difference between the body coordinate system and the global coordinate system. This is showing the difference resulting from the equilibrium roll and pitch.

The next command needs to be looked at a bit closely. Here we are asking for a report of the status of the pressure on the defined holes, *&status p_hole*. Please note that this is the static situation at the current event. All of the holes are going to show that their current state is open. Their current state is in a static scenario. We have not told MOSES that we will be flooding dynamically, nor that this is to be associated with an event.

The command *&status v_hole* produces a status of the valve data. The valve data reported is the hole type, the location, the normal, the friction factor, and the area.

Please notice that the coordinate system being used here is our "new" coordinate system that was defined with the *&describe part* command.

From this point forward, it should be similar to that which we discussed on sink.cif. In the next command, we perturb the system a slight bit and report the position of the bottom of two of the legs. We define an environment so that we can have a time to work with. Then, we open the valve and see what happens. Even in the output we report many of the same quantities.

After you run it you will see in the log file that the simulation ended due to capsizing. If you look at the movie, you will see that the jacket stays with a face near the waterline until near 65 seconds. Then, the jacket begins to roll until it rolls past 90 degrees and the program shuts it down.

Exercise A

Restart sink analysis and at the bottom window type the commands:

```
prcpost
 trajectory
 vlist
```

Hit "ENTER" at the end of each command. Note the title bar changes to show what menu you are in.
After entering the command *vlist*, we get a list with 24 entries. From the resulting list, we know 1 corresponds to time and 7 corresponds to Z:TBRG. Z:TBRG is understood to be the Z location of the barge coordinate system origin measured in the global coordinate system. This helps you understand the numbers after the *plot* command. In the CIF file, the "-no" option is used with the *plot* command. The "-no" option tells MOSES that there are no editing changes to the plot that is to be produced.

1. What is associated with 9?
2. When you look at the plot in the sink.ans directory (gra00001.png).
    - What is the legend on the x–axis?
    - What is the legend on the left hand y–axis?
    - What is the legend on the right hand y–axis (RAX)?
3. When you review the *plot* command in the manual, what command do you need to make a plot with the independent variable to be labeled with "EVENT," the left hand axis "Displ:TBRG" and the right hand y-axis to be labeled with "Bot. Clear:TBRG?"

Exercise B

The sink command file contains a section for reporting the tank flooding.

```
tank_fl
 report
end
```

Add this section to the up_damage command file. When you review the output file for both sink and up_damage you will notice that the columns for Extrnal Fl. Head, Internal. Fl. Head, and Vlv Diff Head are blank for the active compartment. This is caused by the time domain ending abruptly. When the time domain ends due to capsizing, MOSES interprets this as a failed analysis and tries to bring attention to it. There is a message in the log file and this blank in the report that specifically show what is happening with the dynamically flooded compartments.

In order to avoid this, the time allowed needs to change to something smaller than when it capsizes. In the sink command file, change the environment command so that the time simulation ends at 200 seconds. In the up_damage command file, change the environment command so that the time simulation ends at 80 seconds.

&env null –time 200 1

1. Is there a change in the log file?
2. Is there a change in the Compartment Flooding Report?

Exercise C

In the up_damage command file, change the flooded leg to be leg 2, the one associated with compartment "two." From the locations of the valves you should be able to determine that leg 2 is the leg not at the water surface. When this leg floods, the trajectory looks more acceptable to a controlled upend procedure.

4.2 Basic Frequency Domain Motion

Topics:

- Calculating RAOs
- Reporting motions at a point

Reference files:

/ultra/hdesk/runs/samples/sea_keep/cargo.cif cargo.dat
/ultra/hdesk/runs/samples/sea_keep/rao.cif
/ultra/hdesk/runs/samples/data/pcomp.dat

Discussion

The command file for cargo.cif is very similar to rao.cif.

The files rao.cif and rao.dat are discussed on the web page:

http://bentley.ultramarine.com/hdesk/runs/samples/sea_keep/rao.htm

It is a good idea to read the entire section on sea keeping that is available on the web:

http://bentley.ultramarine.com/hdesk/runs/samples/sea_keep/doc.htm

The file cargo.cif is in the /ultra/hdesk/samples/sea_keep directory. This CIF file shows how to do a frequency domain motion analysis. This is the first of two exercises that deals with motion analysis. For this first exercise, we introduce the concept of parts. In the next exercise, the parts will be held together with connectors. Let's start by looking at the pcomp.dat file. In the Basic Stability exercise, we used a part but it was not discussed. In this file, we are again using the cargo barge cbrg180 from the library. The new part is under the section labeled "add cargo." This section begins with the command *&describe part cargo*. In MOSES, a part is defined within the body coordinate system. Here the coordinate system references the barge origin. Please recall that for the vessels in the vessel library, the origin of the coordinate system is the intersection of the bow, the centerline, and keel. The command *&describe part cargo* simply tells MOSES that we are going to describe a part and that MOSES should classify it appropriately. A body can have many parts and each part can have its own name. Parts with the same name as the body are referred to as "body parts." It is the intent to show through example how to use body parts and how to use "regular parts."

The part that we are going to describe will have a weight and a piece associated with it. The weight is used to define the mass matrix and the piece is used to define the wind and current areas to attract wind and current loads, respectively. It is not necessary to define the wind and current areas in this manner. As an example, we

are showing how to define the wind and current areas by employing a piece.

Let's discuss the commands **, #weight, pgen, plane* and *end_pgen*. The first item is defining the point *car_cg. We need to somehow describe the geometry of our items (cargo) to MOSES. We describe the geometry by a set of points. Then, we tell MOSES what we want at the points or if we want to join the points to make a surface. Here we are describing the point where the center of gravity of the cargo will be.

We define the weight of the cargo with the command *#weight*. (It is assumed that by this point the student can look up the command format for *#weight* in the online manual).

To define the cargo, we need the mass properties and the geometry description. The geometry description is essentially the piece description. In this case, the piece is generated with *pgen*, piece generator. The commands *pgen, plane* and *end_pgen* describe the geometry. Here we do not define points. We are describing the geometry much like a ship plan and let MOSES generate the needed points. The numbers 65 to 115 tell MOSES the station locations. Remember, these stations or *PLANE* are measured from the bow. The first plane will be at an X location of 65 feet from the bow. The option *–rect* describes the station properties. There is only one station property so that all of the stations will be the same. All of the stations will have a rectangular shape with the bottom of the rectangle at Z = 15 ft, the top of the rectangle at Z = 30 ft, and the total beam 66 ft. If you look up the format for the option *–rect*, you will see it asks for a ZTOP, ZBOT, and a BEAM. In conclusion, we have a cargo shaped like a box with the mass properties of 1000 kips, radii of gyration Kxx = 16 ft, Kyy = 16 ft, and Kzz = 20 ft.

Now, let's start talking about the CIF file. The first part of the CIF file we are already familiar with. We know how to put the barge at the desired draft and trim, we know how to put ballast in a compartment, and we know how to review our setup with the *&status* command and options. We have familiarized ourselves with the part designated as "stability trans" in the first three exercises.

We are going to discuss the general trend of the CIF file here in the workbook since the commands in the CIF file can be explained by following the web page discussion on rao.cif.

Generating the hydrodynamic database is one part of the analysis that can take much computer time. Strip theory in general is faster than 3D–Diffraction. The greater the number of panels used to define a diffraction mesh, the greater the computer time needed to generate a hydrodynamic database.

In the frequency response menu, most of the commands begin with the "fr" or "st" characters. The commands that begin with "fr" compute frequency responses and the commands that begin with "st" compute the statistics. What comes after the underscore tells MOSES what quantity we are looking for. Here, we used "point"

when we are interested in the frequency response at a point. Please note that there is always an "fr" command (the responses have to be computed first) before any statistics can be computed and reported.

Exercise A

Change the file cargo.dat so that line 11 reads:

```
pgen cargo $  -perm 0 -cs_cur 1 1 1 -cs_win 1 1 0
```

Compare the changes in righting arm results this change causes. If you look up the *pgen* command in the manual, do the changes make sense?

Exercise B

The project has informed you that the transportation will now have two pieces of cargo. Leave the current cargo where it is and add the second one with the following information:

- The dimensions of the cargo: length 50 ft, width 66ft, depth 15 ft.
- The CG is x = 142, y = 0, z = 20.
- The cargo weight is 1000 kips

You are to add the cargo and empty compartments 4p and 4s. You are asked for an updated stability and RAOs.

This is the suggested addition to the DAT file. The answers in the answers section are based on these changes:

```
&describe part cr_strn
*cr_strn 0 0 20
#WEIGHT *cr_strn 1000 16 16 20
pgen cr_strn -perm 0 -cs_curr 1 1 1 -cs_win 1 1 0
plane -25 -20 -15 -10 0 10 15 20 25 -rect 15 30 66
end_pgen
```

Changes to the command file.

```
INMODEL
medit
&describe body cbrg180
&describe part cr_strn -move 142 0 0 0 0 0
end_medit

&compartment -percent cbrg180 @0 \
3p 100 1.0255 3s 100 1.0255 \
1p 100 1.0255 1s 100 1.0255
```

1. Did the range change in the stability results? What is the range?
2. When you present the results, are you going to make a comment about the resulting condition (draft, roll, trim)?

5 Connectors

Connectors are mainly used to transmit a force or moment from one body to another. The amount of force or moment is dependent on the connector definition. In this section we present a variety of connector types and a variety of uses.

5.1 Sling Assemblies

Topics:

- Introduction to sling assemblies
- Defining macros
- Working with more than one body

Reference files: two_blk.cif, two_blk.dat, up_lower.cif, up_lower.dat, spread.cif spread.dat

The presentation here is meant to show a progression of connectors and bodies. The first set of files shows one set of connectors between a hook (ground body) and a jacket body. The second set of files show two sets of connectors between two separate hooks and one tripod jacket body. The third set of files shows one set of connectors between a hook and a spreader bar, a second set of connectors between the spreader bar and a deck, then separately a third set of connectors between a hook and the deck.

File Name	Analysis	No. Bodies	No. Tip Hook
up_lower	Lower	1	1
	Upend	1	1
two_blk	Lower	1	1
	Upend	1	2
spread	Lift	2	2

Table 4: Pregessive Complexity of Exercise

For both the up_lower and the two_blk analysis two processes are defined. One process for the lowering and one process for the upending. These files also show how to activate connectors when in use and deactivate connectors when not in use.

Discussion: Single Tip Hook Assembly

This set of files shows how to define a single tip hook assembly. The discussion for up_lowr.cif are located at:

http://bentley.ultramarine.com/hdesk/runs/samples/install/up_lowr.htm

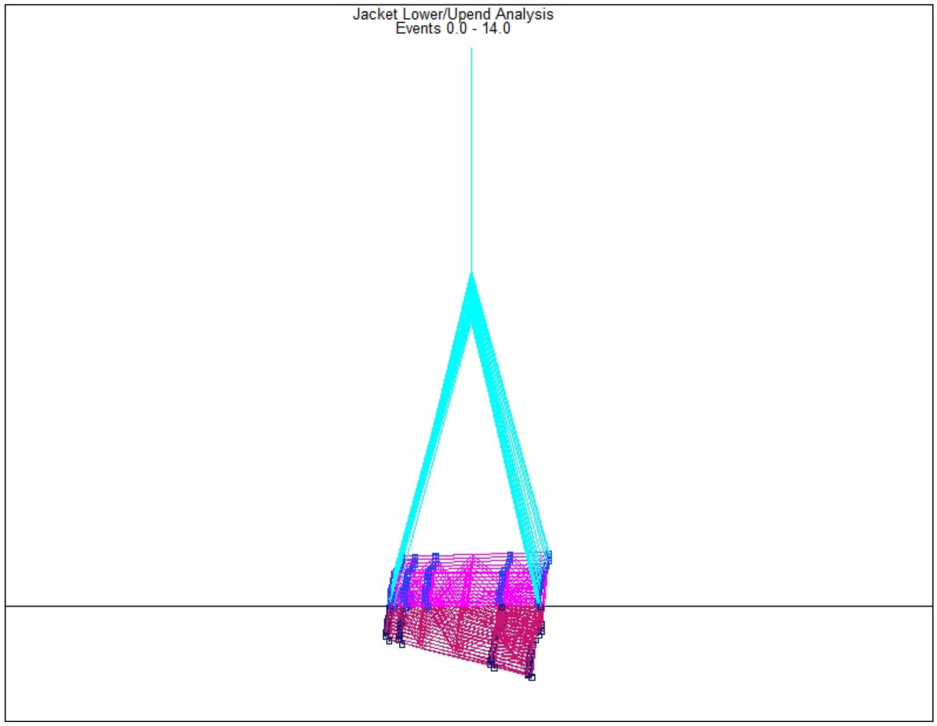


Figure 29: Side view of 4 pile jacket during lowering process

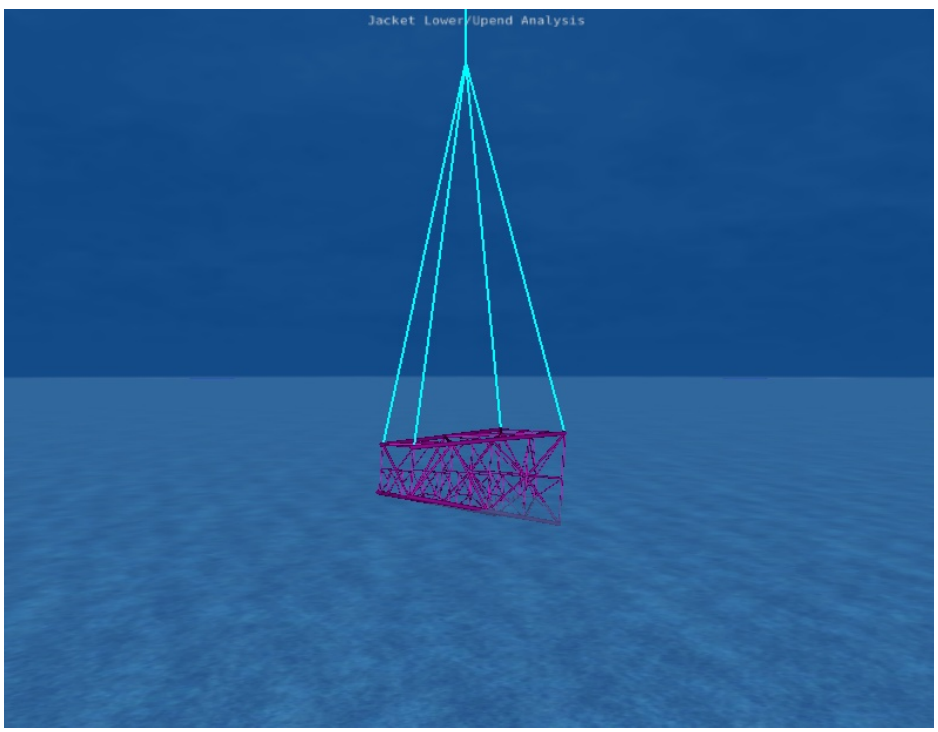


Figure 30: Isometric view of 4 pile jacket during lowering process

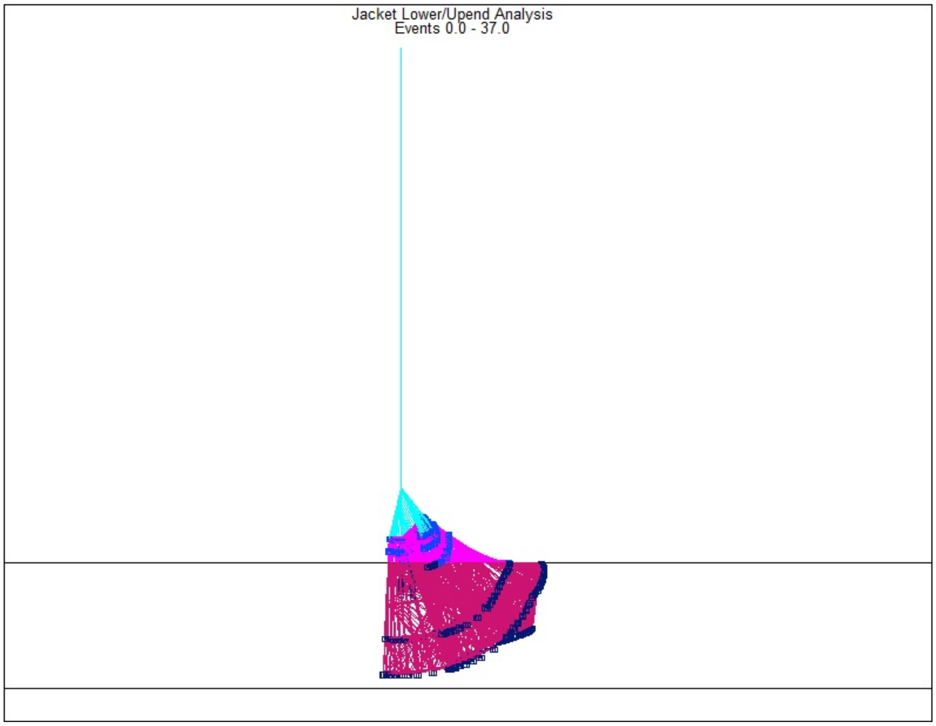


Figure 31: Upending with a single Tip Hook Assembly

Discussion: Tip Hook Assembly Using Two Separate Hooks

This set of files shows how to use a sling assembly to lower a jacket into the water. Then a new sling assembly is defined and used for upending.

A similar analysis is done in the two_blk files found at:

http://bentley.ultramarine.com/hdesk/runs/samples/install/two_blk.htm

Here a single hook and assembly set is used to analyse the lowering of a tripod into the water. The second processes uses two hooks to upend the tripod.

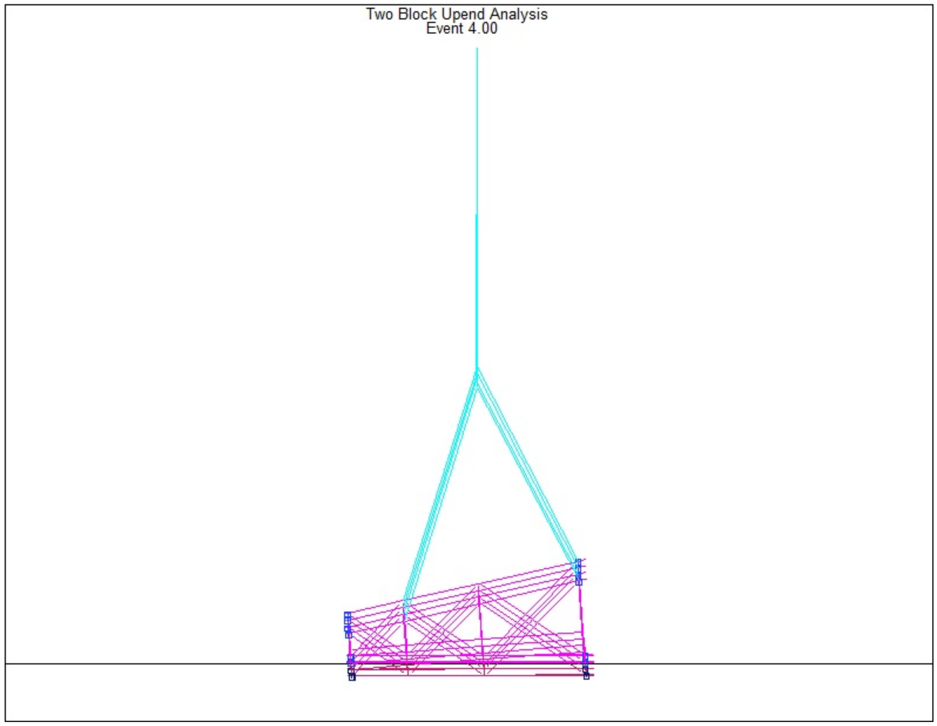


Figure 32: Side view of tripod jacket during lowering process

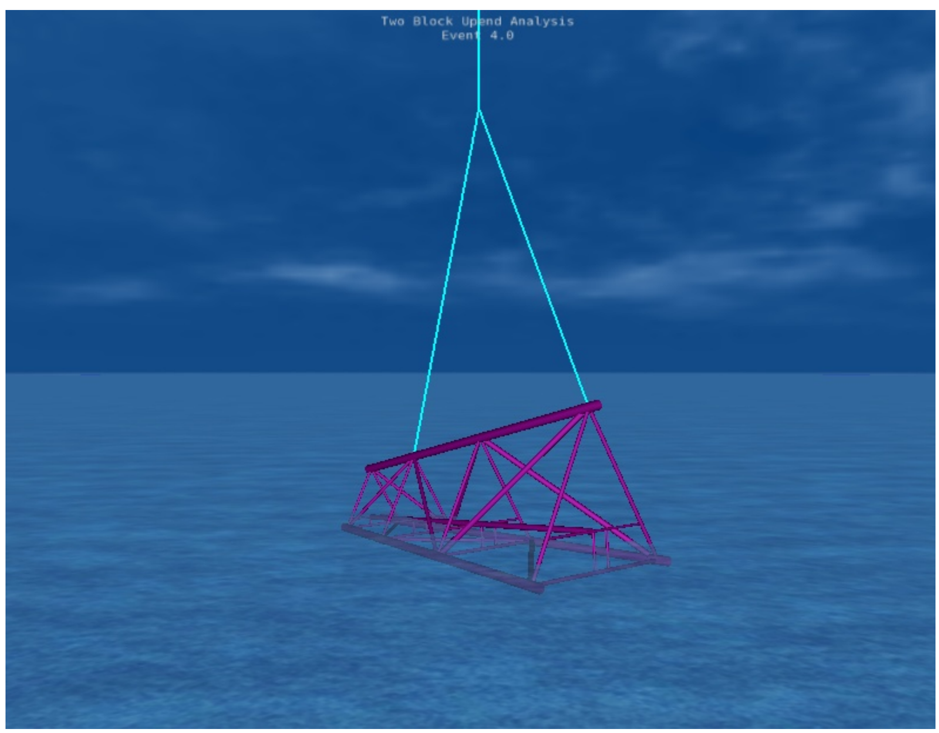


Figure 33: Isometric view of tripod jacket during lowering process

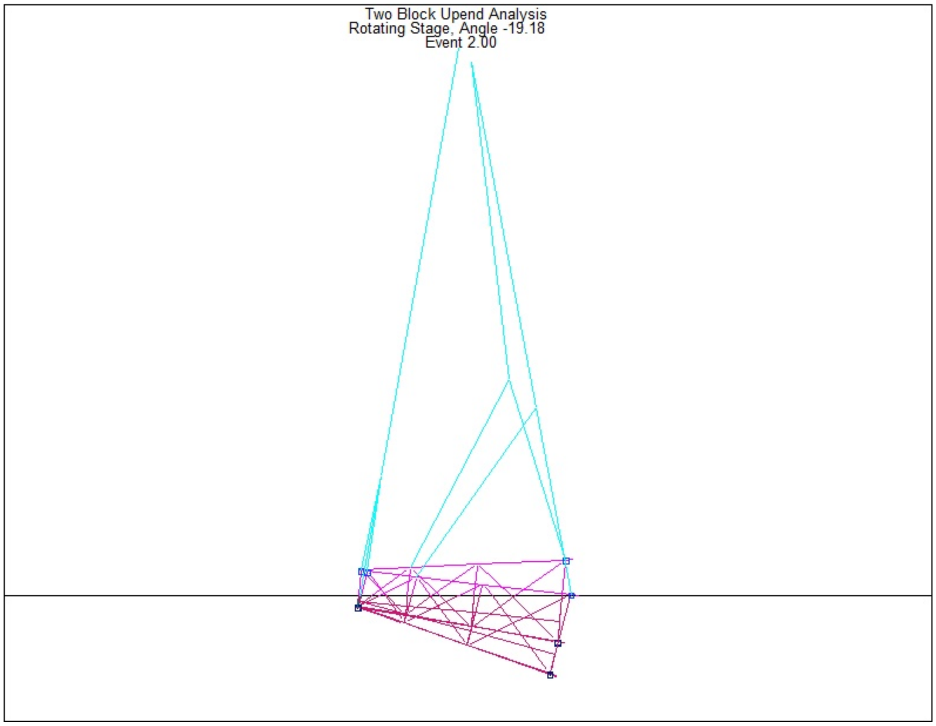


Figure 34: Upending with a Two Hook Assemblies

Discussion: Tip Hook Assembly Using Two Separate Hooks and a Spreader Bar

Finally the spread files show how to define a spreader bar for use in lifts. The list and trim is because center of gravity is not a tthe gemoetric center. The results of the command *&status b_w* also show this.

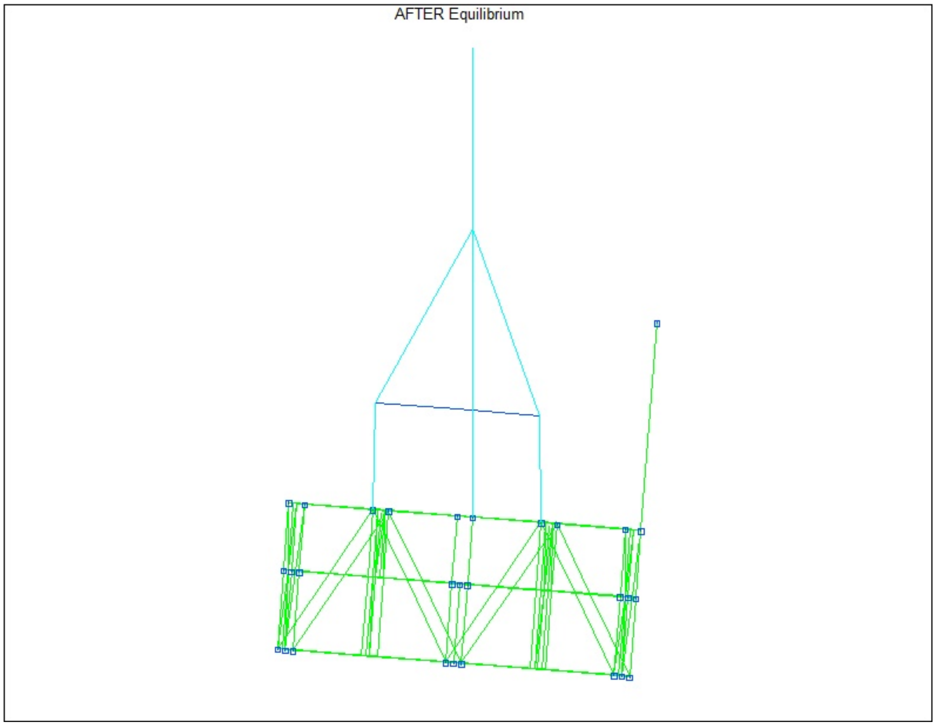


Figure 35: Front view of deck lift

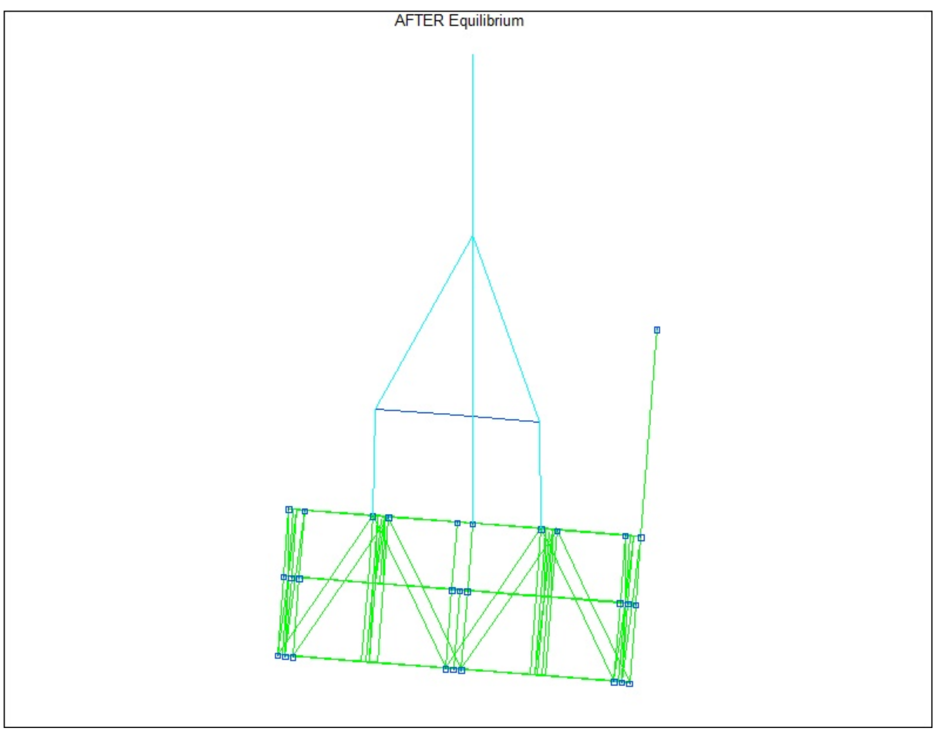


Figure 36: Side view of deck lift

Exercise A - up_lowr files

Does the restraint report for event 1 of the lowering process agree with the connector force magnitude report?

Exercise B - up_lowr files

The environment description inculdes the option “-mean yes”. What changes in the Connector Force Statistics report when this is changed to “-mean no”.

5.2 Modeling a Fender

Topics:

- Working with the Model Edit menu
- Working with Generalized Springs

Reference files

fender.cif fender.dat

These files show how to use generalized springs to model the connection between two bodies. The two barges are positioned along side each other (the centerline axes are parallel). Generalized springs are used to define the fenders between the two vessels. A short command sequence to test the fender definition is presented.

The data file is two rectangular bodies. The user should be familiar enough with the modeling language so as to not need this file explained. Please refer the stability check exercise presented the modeling language if you are unfamiliar with the language presented in the fender.dat file.

Discussion fender.cif

This is the first time we set the location of two bodies with one command. You can see that the *&instate* command is used with two *-locate options*. The barge beams are 15.85 meters for body barge and 27.3 meters for body tanker. The minimum distance between the two is 43.21 (15.85+27.3) meters. The instate command leaves the centerline of the body tanker at x = 0 and y = 0 in the global coordinate system and leaves the centerline of the body barge at x = 180, y = 45.2 m in the global coordinate system. This definition leaves a 2 meter distance for the fenders. This places the barge port side shell at y = 29.35.

For most commands, it is acceptable to have the list of multiple options. The same can be said about the *&weight* command, which follows. These two commands have been discussed in previous exercises. The only addition here is that we are using them to define properties of two bodies instead of just one. Please note that this analysis progresses only through the static analysis, therefore it is acceptable for the values for the radii of gyration to be left at 1.

The generalized springs (∼GSPR) are connectors which means we need to add to the model. Therefore, we need to enter the model edit menu. We enter the model edit menu with the command *MEDIT*.

As you may have guessed, some of the commands in the model edit menu that we have thus far used in the data files will be used here to add to the model. The first command, which is familiar to us, is *&describe body*. To define the points we use

the $*$.

This file also comes with notes imbedded within the file itself. The first set of notes address some assumptions about how the fenders are going to work. Here are the contents of the note:

```
$ NOTE:
$ Remember, when defining locations for fenders (compression only
$ gsprs), no force will be generated until the 2 nodes are touching
$
$ Fenders should be located at the waterline alongside the ship, so the
$ point defined should be outboard the vessel by the diameter of the
$ fender
$
$ Assume the fender is 2.05 meters in diameter, and B/2 is 27.3 so the
$ location of the fender point will be 27.3+2.05=29.35
$
$ For demonstration, I'm making a simple assumption that the tanker is
$ wall-sided. This may not be true.
$
```

This is a good time to point out that the power of MOSES is in its ability to be programmed. Part of the burden of programming is leaving enough documentation so that other people, or even yourself after a few months, can easily return to the command file and use it with minimal effort.

We do need to address syntax here. In MOSES, the comment character is $. MOSES will ignore all of the characters in the command after it reads the $ character. Please be aware that some commands can be structured so that they occupy several lines of text, but when you view them in terms of a command item, the lines are a continuation of the command. For example, the *&instate* command we used earlier to define the location of both bodies occupies two lines. A comment after the \ would tell MOSES to ignore the location option for the barge.

The note is telling us that the fender attachment point on the body tanker is located at y = 29.35, or 29.35m on the starboard side. This locates them at y = 29.35 in the global coordinate system. These are all of the points that begin with the five characters $*$fent. In the next section, we find the attachment points on the barge. These are all of the points that begin with the five characters $*$fenb. The note is telling us that the fender attachment point on the body barge is located at y = -15.85, or 15.85m on the port side. This locates the barge attachment point at y = 29.35 in the global coordinate system.

If you take into account the *x* location of the body barge, you will see that the four attachment points on the barge are defined at the same global location as the four attachment points on the tanker.

This leads us into the connector definition. Defining a connector consists of two steps:

first, you define the class, then you define the connector. For the fenders, we are going to be using generalized springs which are listed under the Flexible Connector Classes. The manual page on flexible connectors can be found at the following link:

http://bentley.ultramarine.com/hdesk/ref_man/cls_flx.htm

The format and the use of the command that we are working with is:

~CLASS, GSPR, SENSE, DF(1), SPV(1), AF(1) . . . \
DF(n), SPV(n), AF(n),

```
~fend GSPR compression x 100 2000 y 1 2000 z 1 2000
```

The name of the class is ~fend. The ~ is part of the name. We are defining a compression element with a spring constant K = 100 $mtons/m$ in the element x direction. The maximum allowable force is 2000 mtons. In the element y and z, direction has a spring constant K = 1 $mton/m$ and the maximum allowable force is 2000 mtons. Basically, the element has the x direction as the strong axis, which means that we need to be clear about what the element x direction is.

For the definition, we need to look a the *CONNECTOR* command.

http://bentley.ultramarine.com/hdesk/ref_man/conn_rest.htm#EULER

By default, the element system is aligned with the body system of the body to which the first node belongs. The use of the *-EULER* option changes the element system. For our setup, the first nodes are the nodes associated with the body tanker. For the body tanker, the x is defined bow to stern, the y is defined port to starboard, and the z is up from the keel. The origin is the intersection of bow, centerline, and keel.

In the definition of our fender *~GSPR* connectors, we are using the option *-euler 0 0 90*, which means that the element x direction will be parallel and in the same direction as the body system y direction. The notes within the command file also explain the system change.

This concludes the connector definition. Therefore, we exit out of the Model Edit menu with the *END_MEDIT* command.

The remainder of the file tests our setup. The first report is a geometry report, *&status g_connector*, to check the connection locations. This report tabulates the connection points for each body in each body coordinate system. This report presents the location of the ∗fentX and ∗fenbX in the local body coordinate system.

In the rest of the commands, we move the barge along the global y axis and report the forces. The force should increase when the distance between the barge and the tanker is less than 2.05m, meaning that the connector is in compression. The force should remain at zero when the distance between the barge and the tanker is larger than 2.05m, meaning that the connector is in tension. Since we have defined our

GSPR to be a compression element only, then it should be turned off for any positive forces.

For all of these moves, we are going to be using the F = kx basic equation in the initial position of x = 0. The force reported with the *&status f_connect* command shows that the force in each fender is also zero. Every move is done in a set of four lines. The results of the four line command are all placed in the log file. They are all similar to the first set. Therefore, only the first set will be discussed in detail.

```
&type
&type connector force - barge 1m towards tanker
&instate -move barge 0 -1 0
&status f_connector
```

The first *&type* writes a blank line. The second *&type* command leaves a short message in the log file. The purpose of the message is to make the log file easier to read. If a group of tables are presented one right after another and the only thing distinguishing them is the values, it helps to leave a short message to keep track of why the values are different. The *&instate* command moves the barge in the global y–axis. Some of the moves are in the negative direction, some of the moves are in the positive direction. The last command *&status f_connector* reports the forces in the connectors. For our case, it reports the forces in the fenders.

The first move decreases the distance between the two bodies. The barge is moved towards the tanker by 1 meter. The forces on the connectors report show a force of 100 mtons. For the second move, it again decreases the distance between the bodies. In the last three moves, it pulls the barge away from the tanker. The last two positions report 0 mtons force in the connectors. The second to last position is again for x = 0, the last position the connectors would be in tension.

The command file is exited with the *&fini* command.

Exercise A

Change the orientation of the barge such that the centerline of the barge is perpendicular to the tanker centerline. Place the intersection of the barge centerline and bow at the tankers amidships, keeping the 2.05m spacing for the fenders. Keep the draft for both vessels the same as in the original files.

The four fender connectors are going to have to fit within the breadth of the barge at the bow. Space the fenders with two on the port side, two on the starboard side. Place them 5 and 10 meters from the barge centerline.

You should be able to get the same compression forces as we had with the original files. Remember that compression in the tanker coordinate system is going to be a positive *x* force, so the use of *-euler 0 0 90* stays.

5.3 Basic Mooring

Topics:

- Introduction to connectors (flexible)
- Introduction to Linear Frequency Analysis
- Introduction to Spectral Frequency Analysis
- Introduction to Time Domain Analysis
- Demonstrate some graphic interactive features

Reference files

mp_moor.cif mp_moor.dat

Modeling Discussion

For a general discussion of mooring analyses performed with MOSES please see:

http://bentley.ultramarine.com/hdesk/runs/samples/mooring/doc.htm

This file shows many MOSES capabilities. This file is also discussed in detail on the website:

http://bentley.ultramarine.com/hdesk/runs/samples/mooring/mp_moor.htm

The workbook is a complement to the material presented on the website; it is not a substitute. The material presented on the website covers the same file, but there are different aspects presented here.

First, we discuss the modeling. In the next exercise, we discuss the commands to perform a dynamic analysis.

The reference files are in the samples directory under "mooring." The reader should be familiar enough with the web page to be able to locate these files and place them in the directory they will be using.

We will start by discussing the DAT file since it is rather short. The first new options we see are the *–cs_wind* and the *–cs_curr* being used with the command *pgen*. The reader should be familiar enough with the manual to find the manual page corresponding to *pgen*. Basically, we are telling MOSES to compute the exposed wind and current area, and then compute the force based on wind and current speed. If you would like a more detailed discussion, please see Wind and Current Force of

the verification document located at the following link:

http://bentley.ultramarine.com/hdesk/document/include/verify.pdf

That covers the new parts of the DAT file. Now for the CIF file.

The first new part is the section that begins with *medit* and ends with *END_MEDIT*. *Medit* stands for "model edit." So, we are going to alter the model. For this exercise, there is only one body, TBRG, in our model. When we edit the model, MOSES will default to TBRG. If we had more than one body, we would have had to use the *&describe body* command to tell MOSES which model we were editing.

The first thing we do is define four nodes *MLA, *MLB, *MLC, and *MLD. If you look at the coordinates and look at the DAT file, you will notice that the four coordinates are at the four corners of the barge. These four points will be the fairlead locations (where the mooring lines are attached to the vessel).

The next line that begins with a ~ defines the class. It is important that we understand the concept of class. This is how MOSES associates elements with properties. Please review the entire section at:

http://bentley.ultramarine.com/hdesk/ref_man/cls.htm

I stress that the entire section be reviewed because there are many types of classes. However, we will only be dealing with flexible classes in this exercise.

We are going to use the flexible type b_cat. The type of b_cat connects a body to the ground, or as I like to think, it connects the body to the bottom.

Also, I want to clarify that 4000 ft is the starting length for MOSES. We may ask MOSES to change the length depending on the tension, horizontal force, or another parameter. The 4000 ft length can be changed, much as in real life, the mooring line length can be let out or drawn in.

Now we define the actual lines; a through h. There are two lines at each corner. Here as with everywhere else, if you do not name an item, MOSES will provide a name. Instead of having MOSES make up a name, we are using letters to name them. You are free to name them whatever you like. There is an eight character limit.

That is the end of editing the model. All of these commands could have been placed in the DAT file. I put them here because of habit. This separates the body model from the connector model. This is because there is normally a need to reposition (as was done in this sample with the *&instate* command) the bodies before the connectors are defined. The repositioning happens in the CIF file, so the connectors are defined after repositioning the body in the CIF file.

The next section is commented with "Move Anchors." Earlier, we let the anchor simply fall into the water so that they landed 20 ft horizontally from the vessel.

Figure 37 shows what the mooring system looks like at this point. The command *&connector @ –a_tension 100* tells MOSES to increase the horizontal distance from the vessel to the anchor and stop when the tension at the fairlead is 100 kips. There are other entities which MOSES can use to stop the change. Please see the *&connector* manual page.

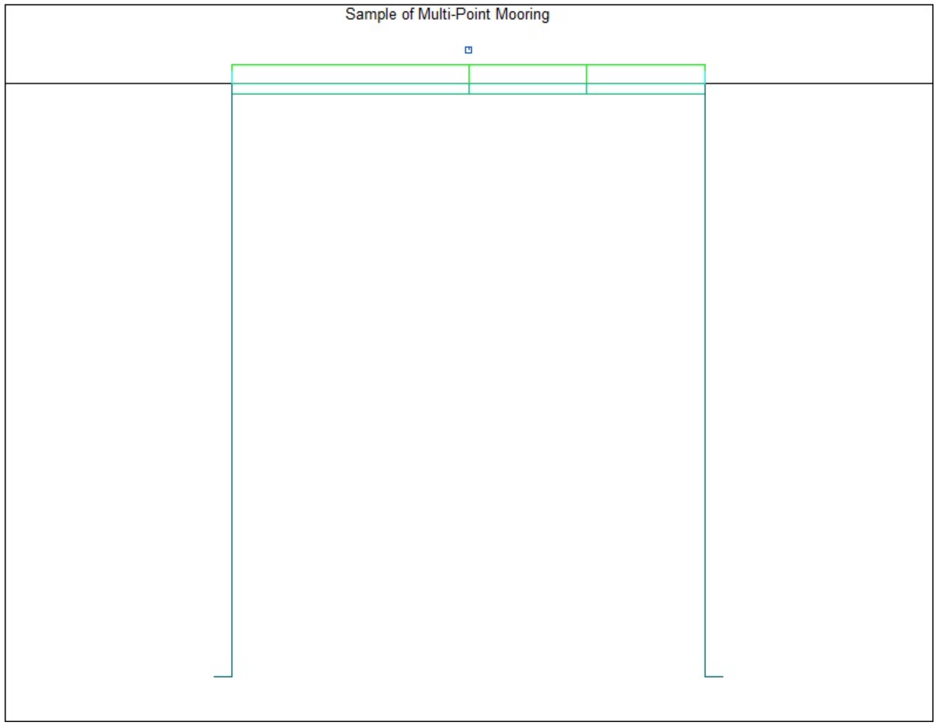


Figure 37: Mooring system before anchor location

Notice here that the length of the line, 4000 ft, is not going to change.

The next command, *&type*, is very handy. Basically, this command is to leave oneself messages in the log file. In the days of slow computers, one would be sitting at the computer wondering if any calculations were happening. If we were to stare at the screen wondering if the computer was working for us, this message would let us know that it is. This is a very basic use. Later we will be asking MOSES to give us more meaningful information about our analysis.

As you may have noticed, we try to make the CIF files a bit easier to read by leaving ourselves messages. The first message was SET BASIC PARAMETERS, and now we have progressed to Mooring Tables. We usually set these messages with a row of *s.

In this section, we are going to produce the characteristics of both the mooring system and a single catenary mooring line. The other thing you may have noticed is that we change the left hand margin depending if we are inside a menu. This is not required;

it just makes it easier to read.

The first command *CONN_DESIGN* enters the Connector Design Menu. Once inside the Connector Design Menu, we are going to create a table of the properties of the catenary mooring line named A. Then, we are going to create a report of the properties of the mooring system when the barge is moved in the 90 deg heading. Please see the manual for the specifics of the command.

This is the end of the modeling part of this example.

This is a good place to do Exercise A.

Preparation for Dynamic Analysis Discussion

Now that we are satisfied with out setup, we still need a hydrodynamic database, environment, and an equilibrium position before any dynamic analysis can be performed.

The next section has been commented as Frequency Domain and was discussed in previous exercises.

Now that we have a hydrodynamic database, let's get an environment definition and place the vessel at an equilibrium position.

The next section has been labeled Find Equilibrium. The first command defines the environment. The second commands tries to find an equilibrium position.

The environment chosen name is TEST. Only a wave environment and time parameters have been defined. Please see the &env command in the manual to see all of the other options available.

I have specifically written "tries to find equilibrium" because there is the possibility that MOSES will not find equilibrium. You need to keep monitoring the log file and if equilibrium is not found, then you need to examine the residuals. For very large bodies, there are times an equilibrium is not found but the analysis can continue.

The reason we find an equilibrium position is that we need the mean static offset position. We need the vessel to be placed in a position where the mooring system forces equal the mean environmental force.

For this sample, at the conclusion of this section the vessel is at the mean static offset position. When you are doing your own mooring analysis, you will need to stop MOSES here and verify that the mean offset position is acceptable.

The next section has been labeled Define Report Points.

Depending on what our task is, we may be interested in all of the points or just a

few points. For this analysis, we are just interested in reporting what happens to the four fairlead points. So, we have told MOSES that the interest points all begin with the two characters *M. The @ sign is the wild character in MOSES. Not only is it the wild character, it also means that any number of characters can be substituted. All of the MOSES special characters are discussed at:

http://bentley.ultramarine.com/hdesk/ref_man/cmd_menu.htm

Dynamic Analysis Discussion

The dynamic analysis portion of the file is separated into the linear frequency domain analysis, the spectral frequency analysis, and the time domain analysis.

Now take out the *&eofile* used for the earlier exercise, and run the entire sample. The discussion will include the CIF, log, and out files.

Dynamic Frequency Domain Analysis Discussion

We are going to discuss the two types of frequency domain analyses here. Notice that the frequency domain has its own menu(*freq_response*). Both the linear frequency domain analysis and the spectral frequency domain analysis are done within this menu.

Let's talk about the Frequency Domain. There is also a discussion on the website at:

http://bentley.ultramarine.com/hdesk/runs/c_htm/frq_com.htm

It is important that you understand the difference between the linear frequency domain analysis and the spectral response analysis. We are not going to discuss the theory differences between these two methods. If you would like more discussion on the differences, please read the discussion at the following link:

http://bentley.ultramarine.com/hdesk/ref_man/freq_rsp.htm#RAO

For both the linear frequency domain and the spectral response analysis, we perform the following steps:

- calculate the response
- summarize some of the calculations
- report the frequency response
- report the motion statistics at a point
- report the connector force statistics

The main difference is that for the linear frequency response we use the command *RAO*, and for the spectral response we use the command *sresponse* with many more periods.

In both instances, fr_ is used to report the frequency response. So, *fr* stands for

*f*requency *r*esponse.

In both instances, st_ is used to report the statistics at a point and connector forces. So, *st* stands for *st*atistics. Here, we only reported the motion statistics at a point and connector forces. There are other statistics that can be reported. Please see the manual command index to get a listing.

Notice that each of these commands enter you into the Disposition Menu. MOSES will perform the calculations, but if you do not ask for a *REPORT*, MOSES is not going to give you the results. Also, notice that you need to *END* out of each disposition menu. You need to *END* out of the Frequency Response Menu too.

Dynamic Time Domain Analysis Discussion

Now, let's talk about the Time Domain. There is also a discussion on the website at:

http://bentley.ultramarine.com/hdesk/runs/c_htm/tdom_com.htm

You will notice that the command structure looks very different. The command *TDOM* is a Main Menu command. Please refer to the manual for remarks on the *–NEWMARK* option. If you examine the log file, you will see that there are twenty messages about the database being saved.

Having brought up the subject of databases, this naturally leads into questions about time step choice, total time choice, and many others. Because a time domain simulation can be performed on many configurations, it is difficult to address all of the possible answers one may have on this subject. I would like to refer the reader to the FAQ on Time Domain:

http://bentley.ultramarine.com/hdesk/question/time.htm

In the frequency domain analysis, we had sub–menus to report the results. Here, there is a menu for post processing of processes with events. Since the time domain is essentially a collection of events, we will use the *PRCPOST* menu. We are going to report all of the results within sub–menus of the *PRCPOST* menu.

There are similarities to what we had done before. We will have to issue a *REPORT* command to get the standard report within the Disposition Menu, and we will have to end out of the Disposition Menu.

For the *points* disposition, we have 133 variables. The number of variables here will really depend on how many points you are interested in. For the command *points*, it would be best to always ask for the variable list, *vlist*, then choose the variables you are interested in reporting, then rerun the analysis. For an analysis with more bodies or more mooring lines, you may want to first run the time domain for 10 seconds and get your variables, then run for the proper length of time, and then review your

output.

Depending on the number of bodies in your analysis, you may want to do this for all commands in the *PRCPOST* menu.

There is a new command in the disposition menus under the *PRCPOST* menu. In the frequency domain disposition menus, there were the commands that began with the characters "ST" to report the statistics. In the PostProcess menu, we have the command *STATISTICS*. It is because the number of available variables can change with each analysis that a standard set of column headings for the statistics report was not developed.

Please notice how the *statistics* command uses the numbers listed by the command *vlist*. The Disposition Menu has other ways to present the results. The most popular being *extreme* and *plot*. Please consult the manual for these two commands.

Exercise A

Run the file. MOSES should end in interactive mode.

Type CTRL–G. This should bring up a tab in interactive mode, but the screen will be all blue. This is because MOSES tries to put in all of the mooring system. Type

```
&picture starb -render gl -connector no
```

This creates a picture of only the barge.

```
&picture starb -reset
```

This changes the picture to a wire frame. The picture shows the barge and the mooring system. The following picture is what you should get:

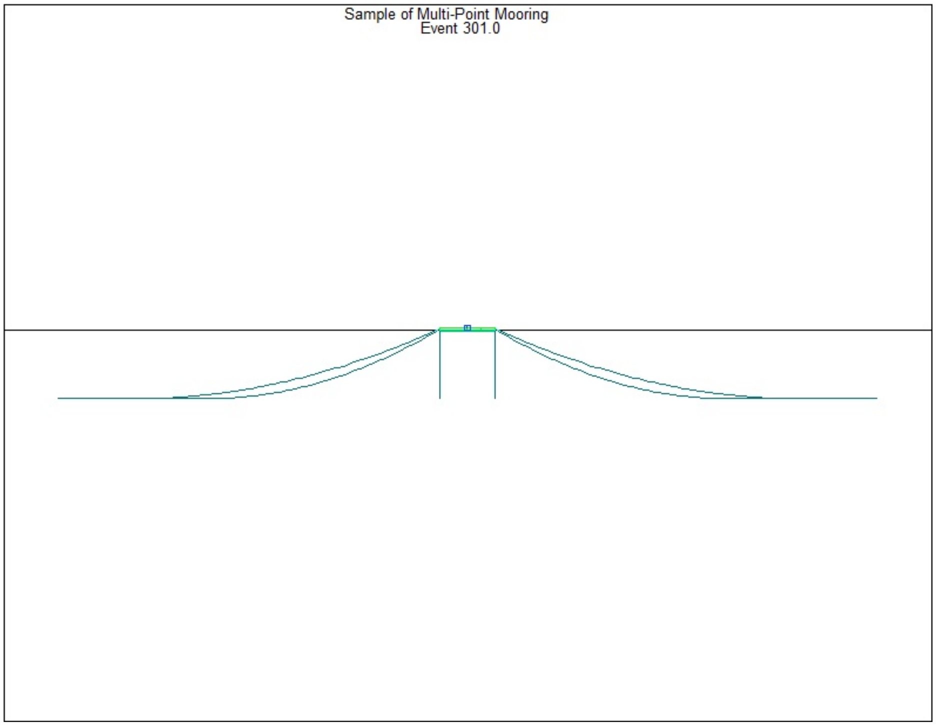


Figure 38: Mooring set up for mp_moor analysis

Type CTRL–F to finish MOSES.

Now, let's test some of the descriptions I have written earlier.

Change mp_moor.cif to read:

```
   CONNECTOR h -anchor 45 20 ~wire *MLA
END
   &device -primary screen
   &eofile
```

Now, get a rendered picture as you did before.

You will see that the mooring lines fall straight to the bottom and that the anchor is placed about 20 ft from the fairlead. Since we have not yet told MOSES to move the anchors, this is exactly what the picture should look like.

Exercise B

Change the anchor location in the bow lines so that the pretension is 70 kips. Change the environment description so that the time step is 0.5 seconds.

1. Make a plot of event vs. FY:TBRG.

2. What is the extreme clearance for point *MLC?

3. What is the mean of the Z location of the barge?
4. Make a plot of event vs. force magnitude on mooring line B.

5.4 CALM Mooring

Topics:

- Connection two floating bodies with a flexible connector
- Equilbrium of separate bodies within the same analysis

Reference files

calm.cif calm.dat

Modeling Discussion

The reference files are in the samples directory under "mooring." The reader should be familiar enough with the web page to be able to locate these files and place them in the directory they will be using. Most of the commands and the arrangement of the commands have been presented earlier.

Here we are going to analyse a tanker connected to a buoy via a hawser.. This would be similar to an offloading scenario. The buoy has the spread mooring system, is attached to the tanker via a hawser. The tanker is allowed to weathervane around the buoy.

This analysis is considered the second part of the mooring analysis exercises.

Command file Discussion

We will start by discussing the data file. Indeed many of the commands are similar. The buoy is represented by a structural model made of beam elements. The vessel model is a panel model created with *pgen*.

The buoy model starts with defining the weight, *#weight*, the added mass *#amass*, and the drag, *#drag* attributes. For the extended, list of attributes please see the following link.

http://bentley.ultramarine.com/hdesk/ref_man/load_g.htm#ELAT

The displacement of the body buoy is defined by a tubular beam element. Two nodes, one at the top (∗bb) and one at the bottom (∗bt) are defined. The command format is the same as the structural model used for the long_str barge beam model, however, different values are used for the options. That concludes the definition of the buoy body.

The barge model is similar to the other vessels we have seen. The attribute *#tanker* is new and the use of the cartesian coordinate system is new. The documentation for the attribute *#tanker* is at the same location as the attributes used in the buoy model. Basically the option *–cart* is followed by y,z pairs. The documentation for

the *–cart* option can be found at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/pieces.htm

That concludes the discussion for the data file. Now, on to the command file.

Command file Discussion

The top of the command files begins as most of the command files have. In previous analysis we have edited the model inside the model edit menu, *medit*. For this analysis we need to be careful and make sure that any model editing is done on the body we specify. This is why there are two *&describe body* commands within the model editing menu.

For the buoy, the command

```
&describe body buoy -md_force 1 0 0
```

is used. This means total mean drift force will not include radiation and coriolis forces. The documentation can be found at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/bod_par.htm

For the buoy we are going to define the 8 fairlead points around the perimeter, and the hawser connector. These are listed at ∗CAT1 to ∗CAT8. The hawser connection is listed as ∗TAUTB. The mooring line description is similar to that used in the files mp_moor. Connectors do not belong to a body. It is a good idea to keep them in the buoy section since they do connect the buoy to ground. The new option *–b_tension* designates the breaking strength. That concludes the buoy model edition.

A similar approach is taken with the tanker model. The same approach for the mean drift force is taken. A node (∗TAUTT) is defined to connect the hawser. Finally the connector between the tanker and the buoy is defined. Again, please remember, that a connector does not belong to a body. It is presented here as part of the tanker description because we needed to define the connection on the tanker before defining the connector.

Here are some views of what the system looks like at this point.

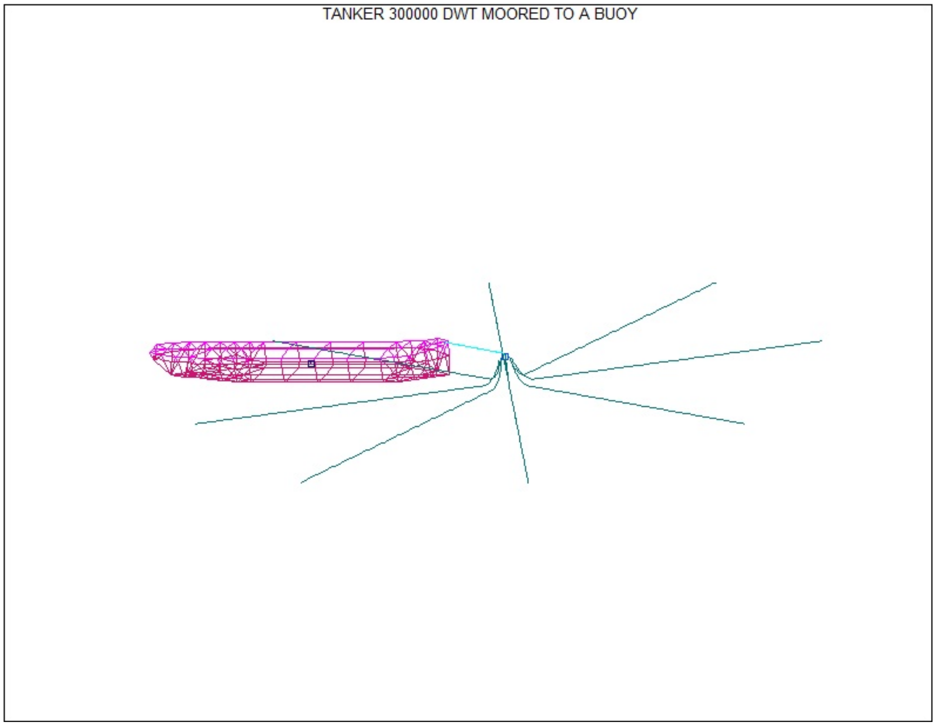


Figure 39: System Isometric View

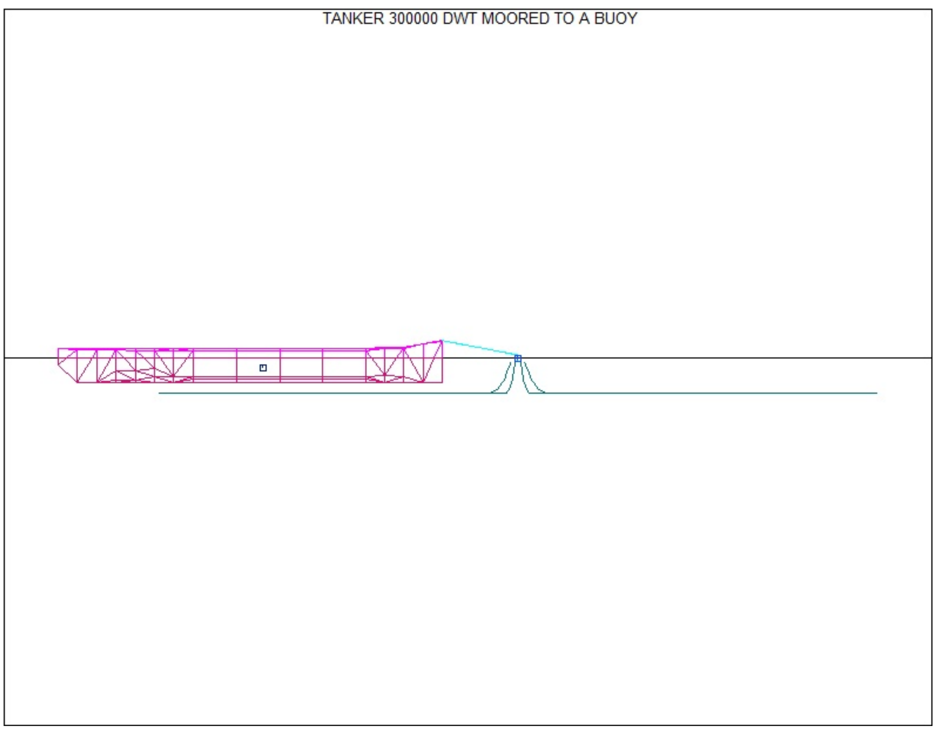


Figure 40: System Side View

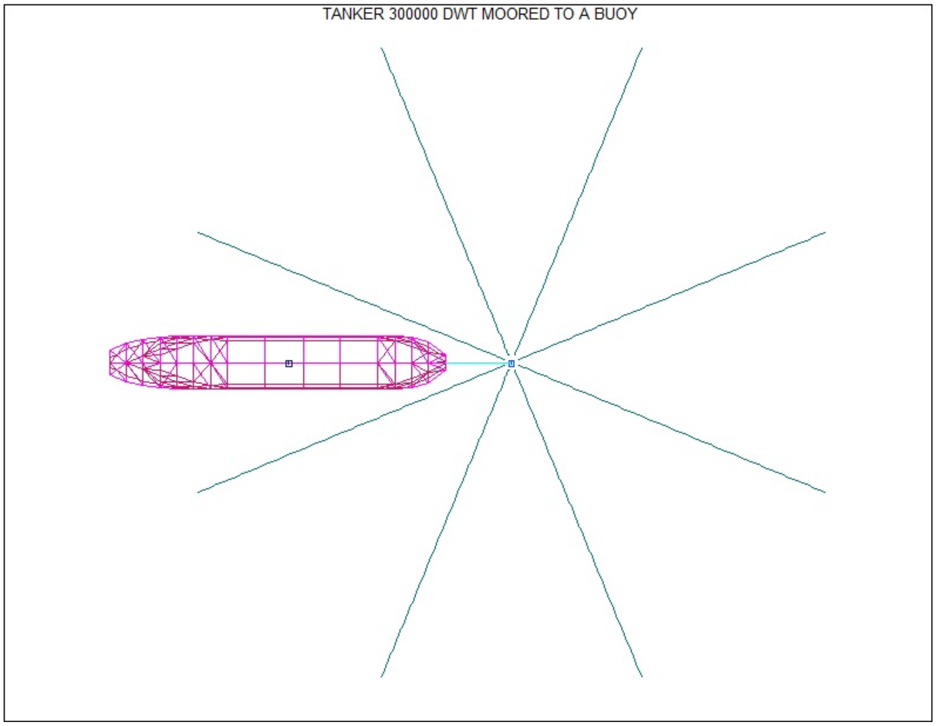


Figure 41: System Top View

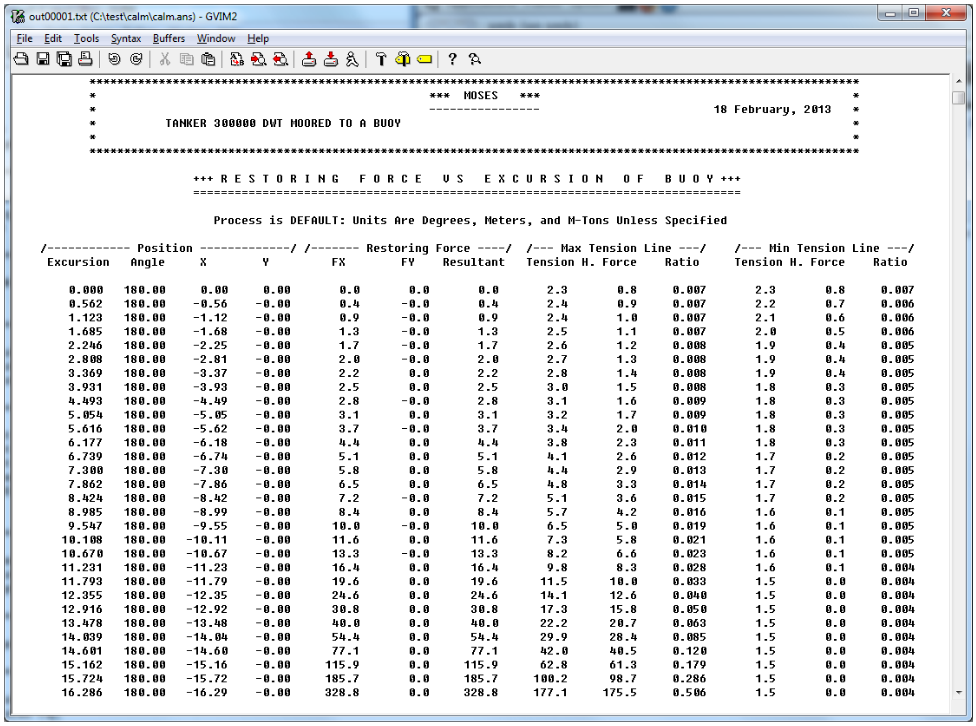
Many of the commands in the sections labeled as “plots” and “set weight” are familiar to us. The new command is

```
&connector taut -INACTIVE
```

This analysis is meant to simulate an offloading scenario. In real life the buoy will sit by itself for a time. In real life the tanker will approach the buoy on its own. This means that each body has to be in equilibrium independent of the other. The command takes out the hawser that we have just defined between the two bodies. The other commands you will be familiar with is altering the mass matrix so that for each body the six degrees of freedom are in equilbrium. The reporting of the categories will show how the mass definition was changed.

At this point it is a good idea to run the analysis and have the log and output files ready for reference.

We have seen the Connector Design *CONN_DESIGN* menu before. In this analysis we added some plots. But essentially it is the same as that in the mp_moor files. We will be using the Restoring Force table created with the *MOVE BUOY 180* command. We will be referring to it by the name ”restoring force table”. Here is a screen shot of the restoring force table.


out00001.txt (C:\test\calm\calm.ans) - GVIM2

File Edit Tools Syntax Buffers Window Help

*** MOSES ***

18 February, 2013

TANKER 300000 DWT MOORED TO A BUOY

+++ R E S T O R I N G   F O R C E   V S   E X C U R S I O N   O F   B U O Y +++

Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified

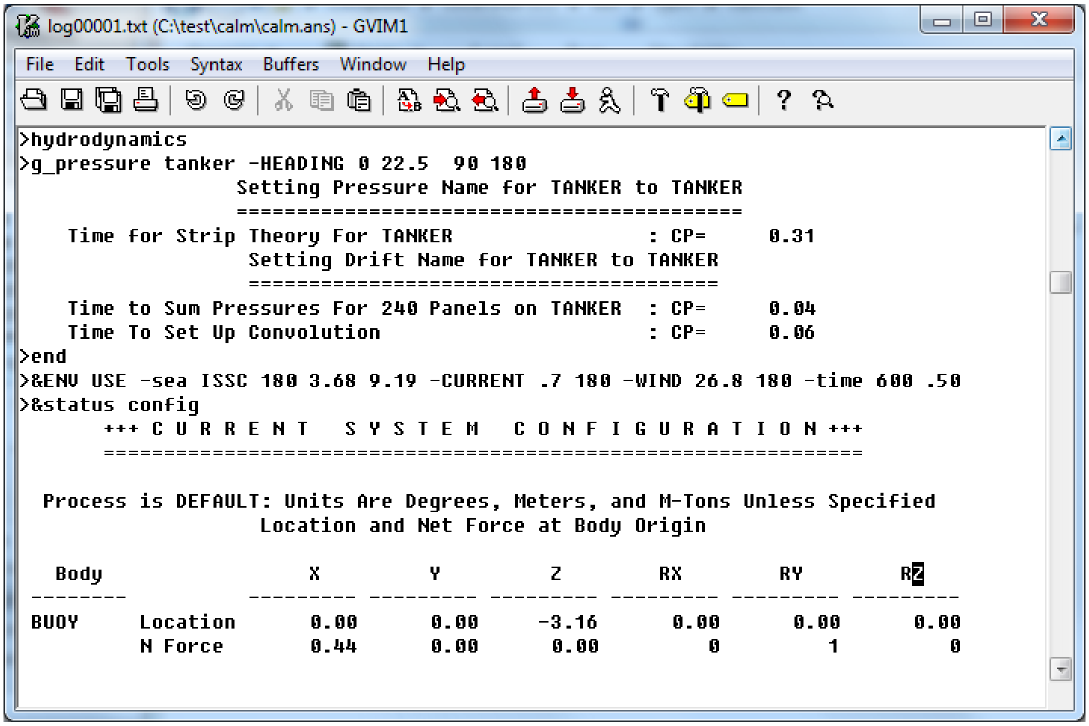
Position				Restoring Force			Max Tension Line			Min Tension Line		
Excursion	Angle	X	Y	FX	FY	Resultant	Tension	H. Force	Ratio	Tension	H. Force	Ratio
0.000	180.00	0.00	0.00	0.0	0.0	0.0	2.3	0.8	0.007	2.3	0.8	0.007
0.562	180.00	-0.56	-0.00	0.4	-0.0	0.4	2.4	0.9	0.007	2.2	0.7	0.006
1.123	180.00	-1.12	-0.00	0.9	-0.0	0.9	2.4	1.0	0.007	2.1	0.6	0.006
1.685	180.00	-1.68	-0.00	1.3	-0.0	1.3	2.5	1.1	0.007	2.0	0.5	0.006
2.246	180.00	-2.25	-0.00	1.7	-0.0	1.7	2.6	1.2	0.008	1.9	0.4	0.005
2.808	180.00	-2.81	-0.00	2.0	-0.0	2.0	2.7	1.3	0.008	1.9	0.4	0.005
3.369	180.00	-3.37	-0.00	2.2	0.0	2.2	2.8	1.4	0.008	1.9	0.4	0.005
3.931	180.00	-3.93	-0.00	2.5	0.0	2.5	3.0	1.5	0.008	1.8	0.3	0.005
4.493	180.00	-4.49	-0.00	2.8	-0.0	2.8	3.1	1.6	0.009	1.8	0.3	0.005
5.054	180.00	-5.05	-0.00	3.1	0.0	3.1	3.2	1.7	0.009	1.8	0.3	0.005
5.616	180.00	-5.62	-0.00	3.7	-0.0	3.7	3.4	2.0	0.010	1.8	0.3	0.005
6.177	180.00	-6.18	-0.00	4.4	0.0	4.4	3.8	2.3	0.011	1.8	0.3	0.005
6.739	180.00	-6.74	-0.00	5.1	0.0	5.1	4.1	2.6	0.012	1.7	0.2	0.005
7.300	180.00	-7.30	-0.00	5.8	0.0	5.8	4.4	2.9	0.013	1.7	0.2	0.005
7.862	180.00	-7.86	-0.00	6.5	0.0	6.5	4.8	3.3	0.014	1.7	0.2	0.005
8.424	180.00	-8.42	-0.00	7.2	-0.0	7.2	5.1	3.6	0.015	1.7	0.2	0.005
8.985	180.00	-8.99	-0.00	8.4	0.0	8.4	5.7	4.2	0.016	1.6	0.1	0.005
9.547	180.00	-9.55	-0.00	10.0	-0.0	10.0	6.5	5.0	0.019	1.6	0.1	0.005
10.108	180.00	-10.11	-0.00	11.6	0.0	11.6	7.3	5.8	0.021	1.6	0.1	0.005
10.670	180.00	-10.67	-0.00	13.3	-0.0	13.3	8.2	6.6	0.023	1.6	0.1	0.005
11.231	180.00	-11.23	-0.00	16.4	0.0	16.4	9.8	8.3	0.028	1.6	0.1	0.004
11.793	180.00	-11.79	-0.00	19.6	0.0	19.6	11.5	10.0	0.033	1.5	0.0	0.004
12.355	180.00	-12.35	-0.00	24.6	0.0	24.6	14.1	12.6	0.040	1.5	0.0	0.004
12.916	180.00	-12.92	-0.00	30.8	0.0	30.8	17.3	15.8	0.050	1.5	0.0	0.004
13.478	180.00	-13.48	-0.00	40.0	0.0	40.0	22.2	20.7	0.063	1.5	0.0	0.004
14.039	180.00	-14.04	-0.00	54.4	0.0	54.4	29.9	28.4	0.085	1.5	0.0	0.004
14.601	180.00	-14.60	-0.00	77.1	0.0	77.1	42.0	40.5	0.120	1.5	0.0	0.004
15.162	180.00	-15.16	-0.00	115.9	0.0	115.9	62.8	61.3	0.179	1.5	0.0	0.004
15.724	180.00	-15.72	-0.00	185.7	0.0	185.7	100.2	98.7	0.286	1.5	0.0	0.004
16.286	180.00	-16.29	-0.00	328.8	0.0	328.8	177.1	175.5	0.506	1.5	0.0	0.004

Figure 42: Force Report

After the connector design menu the hydrodynamic database is calculated. For many of our mooring and flexible connector samples this is the standard approach. First make some checks on the connector system, then continue with the hydrodynamics.

The hydrodynamic database is calculated in the Hydrodynamics menu. Please note that the hydrodynamics of the tanker are the only ones calculated. The body buoy is modeled as a tubular which means that only Morison type forces will be calculated. Figure 42 shows the section of the log file where the hydrodynamic database is calculated. The messages are only about the tanker.

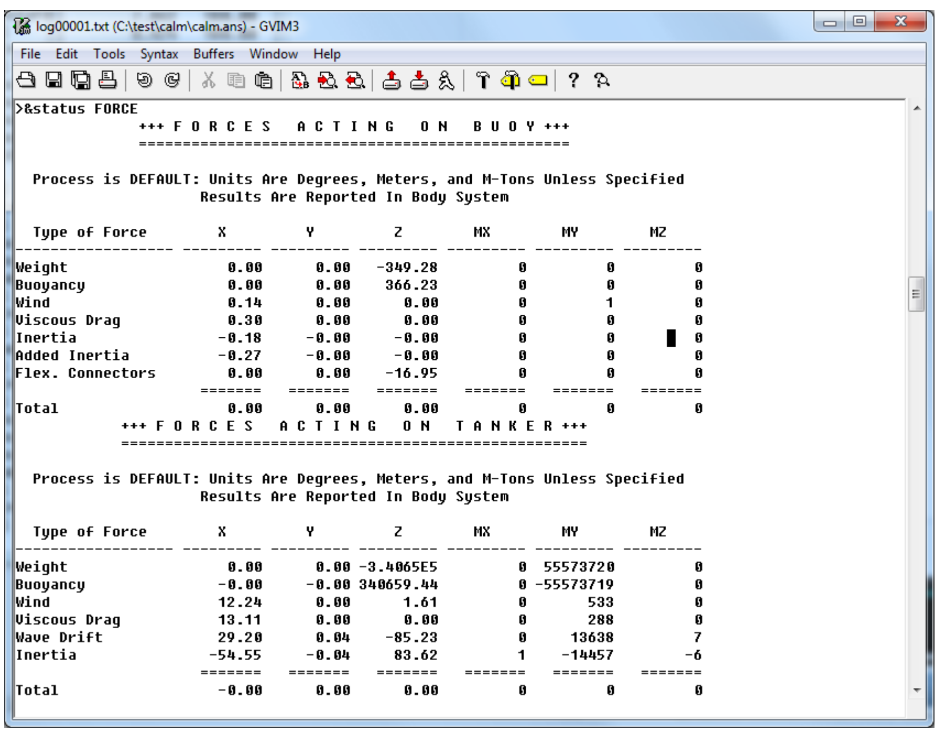
An environment that has waves, current and wind is defined with the command *&env*. This is followed by some *&status* reports.



```
>hydrodynamics
>g_pressure tanker -HEADING 0 22.5  90 180
                  Setting Pressure Name for TANKER to TANKER
                  ==========================================
    Time for Strip Theory For TANKER                : CP=     0.31
                   Setting Drift Name for TANKER to TANKER
                   =======================================
    Time to Sum Pressures For 240 Panels on TANKER  : CP=     0.04
    Time To Set Up Convolution                      : CP=     0.06
>end
>&ENV USE -sea ISSC 180 3.68 9.19 -CURRENT .7 180 -WIND 26.8 180 -time 600 .50
>&status config
       +++ C U R R E N T   S Y S T E M   C O N F I G U R A T I O N +++
       ===============================================================

  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified
                    Location and Net Force at Body Origin

   Body                   X          Y          Z         RX         RY         RZ
 --------              ---------- ---------- ---------- ---------- ---------- ----------
 BUOY     Location       0.00       0.00      -3.16       0.00       0.00       0.00
          N Force        0.44       0.00       0.00          0          1          0
```

Figure 43: Hydrodynamic Calculation Log

Again you will find this approach in many of our samples. A review of the mean forces is reported after the hydrodynamic database has been calculated and the environment has been defined.



```
>&status FORCE
              +++ F O R C E S   A C T I N G   O N   B U O Y +++
              ================================================

  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified
                     Results Are Reported In Body System

  Type of Force          X          Y          Z         MX         MY         MZ
------------------ --------- --------- --------- --------- --------- ---------
Weight                  0.00      0.00   -349.28         0         0         0
Buoyancy                0.00      0.00    366.23         0         0         0
Wind                    0.14      0.00      0.00         0         1         0
Viscous Drag            0.30      0.00      0.00         0         0         0
Inertia                -0.18     -0.00     -0.00         0         0         0
Added Inertia          -0.27     -0.00     -0.00         0         0         0
Flex. Connectors        0.00      0.00    -16.95         0         0         0
                     =======   =======   =======   =======   =======   =======
Total                   0.00      0.00      0.00         0         0         0
            +++ F O R C E S   A C T I N G   O N   T A N K E R +++
            ====================================================

  Process is DEFAULT: Units Are Degrees, Meters, and M-Tons Unless Specified
                     Results Are Reported In Body System

  Type of Force          X          Y          Z         MX         MY         MZ
------------------ --------- --------- --------- --------- --------- ---------
Weight                  0.00      0.00 -3.4065E5         0  55573720         0
Buoyancy               -0.00     -0.00 340659.44         0 -55573719         0
Wind                   12.24      0.00      1.61         0       533         0
Viscous Drag           13.11      0.00      0.00         0       288         0
Wave Drift             29.20      0.04    -85.23         0     13638         7
Inertia               -54.55     -0.04     83.62         1    -14457        -6
                     =======   =======   =======   =======   =======   =======
Total                  -0.00      0.00      0.00         0         0         0
```

Figure 44: Force Report

The next step is to find the mean offset position. We could just start using the *&equi* command until we achieve a position within tolerance. Or we could reposition the two bodies to position that is near equilibrium.

We see that the tanker has a total environmental force of 54.55 m-tons in the positive x direction. This force will be transmitted to the buoy via the hawser. Now, if we look on the Restoring Force table we see that a force of 54.4 m-tons is also the excursion of the buoy 14.039 meters. We know the excursion is not going to be exactly 14.039 meters. There is also a mean force from the buoy that needs to be accounted for. Also the buoy will probably tilt. The section labeled "Set Initial" repositions the two bodies so that they are near equilibrium. You can see how both bodies are moved 13.87 meters in the positive x direction and both bodies are tilted. Also the command

```
&connector TAUT -L_HORIZONTAL %forc
```

sets the tension in the hawser to 51.2 mtons. Which is close to the 54 mtons that is expected.

When the *&equi –omega 1* command is used, equilibrium is found in 5 iterations. This means our repositioning was very close. When we report the configuration, the force, and the force in the connectors we see that we are within tolerance for the equilibrium position. The reports are in the output file. Remember we are looking at the row labeled "Inertia" for both bodies.

Just like the mp_moor analysis three dynamic analyses are performed: linear frequency domain analysis, spectral frequency domain analysis, and time domain analysis.

Most of the reports reference the hawser connections. In two body analysis a great deal of importance is placed on avoiding collisions. This is why the output is concentrating on the positions of ∗TAUTT, ∗TAUTB and the tension in the hawser TAUT.

In the time domain analysis the command *rel_mot* is used. The relative motion between ∗TAUTT and ∗TAUTB and ∗TAUTB and ∗TAUTT is reported. This might at first glance look like we are doing the same thing twice. It is a prudent measure to report the results from each point. The reference for this command is at the following link.

http://bentley.ultramarine.com/hdesk/ref_man/ppo_inte.htm#REL_MOTION

In there we find that:

> The results are expressed in the body system of the first point.

This is important to note. The results are dependent on what coordinate system is being used. For this analysis it is very important to avoid collision. It is a good idea

to look at the minimum distance from several reference points.

Exercise A

What is the minimum distance based on the tanker coordinate system x-y plane?

What is the minimum distance based on the buoy coordinate system x-y plane?

Exercise B

1. Find the location of ∗tautt and ∗tautb at event 1.
2. Find the global position of the buoy and the tanker at event 1.
3. Change the relative motion section so that only the first 4 events are reported

Can you justify with geometry the difference in the two results?

5.5 Sidelift

Topics:

- Tip–hook assembly definition
- Frequency domain motions analysis
- Time domain motions analysis
- Reporting relative distance between two points
- Structural analysis of suspended jacket

Reference files: /ultra/hdesk/runs/samples/how_to/sidelift.cif, sidelift.dat

Overview Discussion

This set of files shows how to perform a sidelift motion and structural analysis with 'native' MOSES commands. There is a boom/hook/sling assembly defined to connect the two bodies, barge and jacket. The jacket is held just above the water.

The two body system is first put in static equilibrium, then the dynamic analysis is performed in both the frequency and time domains.

Many of the steps shown are not necessary to perform the analysis. We use them to show the many options the user has to check the status of the system and evaluate the configuration. The discussion assumes that the reader has the command, data, log, and output file available.

For this exercise, we will:

- Check the motions of the barge and jacket
- Check the jacket for slamming events
- Check the tensions in the boom line and slings. We will assume that the crane capacity is 1000 kips.
- Perform a structural analysis of the jacket at event 5 and the 2nd slam occurrence

Sidelift Data File Discussion

Many of the commands in this file will be familiar to the person that has worked all of the exercises to this point. For the most part, this file is going to be discussed in general terms, with some discussion on the new commands. If there is a command that is unfamiliar, please review the earlier exercises or refer to the reference manual.

Data that we will need for the discussion of the command file are the general dimensions of both bodies. The barge general dimensions are: length = 500 ft, breadth = 170 ft, and depth = 50 ft. The general dimensions of the jacket are: bottom elevation width = 96 ft, top elevation width = 45 ft, and height (from bottom elevation to top elevation) 201 ft.

The top section of this data file has an extra body that is not used for the analysis, but is used for visual guides. This extra body is named ZZZGLOBAXES. For right now, we are just going to acknowledge that it exists. We will talk about it later.

Next, the barge is defined. First, the outer shell is defined; then a crane (using structural elements), and finally some points of interest and a selector.

The last section is the jacket model. This is the same jacket model used in the up_lower sample. Many of the commands used to make the model have been discussed earlier. The last two lines also designate points of interest, this time for the jacket body.

Once the bodies are completely defined, we can begin setting up the analysis.

Sidelift Command File Discussion - Connecting the Two Bodies

The command file starts with the familiar commands that start a command file. The dimensions are set and the model in the data file is read. The command file itself is heavily commented. The discussion here is intended to complement the comments already in the file.

The first new command resets the coordinate system for the jacket part. This is done with command:

```
&describe part jacket -move 0 0 0 *j0501 *j1001 *j0503 *j1003
```

If you read the manual page on the command *&describe part*, you will see that the point order is important for this command. Here we have PT1 (*j0501), PT2 (*j1001), PT3 (*j0503), PT4 (*j1003). The new part x-axis will be from the midpoint connecting PT4 and PT2 to the midpoint connecting PT3 and PT1. The part z-axis is defined by the cross product of the new part x–axis with the vector connecting PT4 to PT2. Finally, the new part y-axis is defined by the new x part axis and the new z part axis and the right hand rule. Resetting the part axis for a part with the same name as the body also resets the body axis. Now that we have reset the part and body coordinate system, the reports that read "Reported in the Jacket Body System" will use this new coordinate system.

We are going to keep referring to joint *J1001 to keep track of the jacket. It would be a more complete analysis if we kept track of the whole lower plane. However, this is an example and we are not going to present that level of detail.

For the points we have designated, the jacket part xy–plane has the face nearest the water. The origin is at the midpoint of the vector between *j1003 and *j1001. The x-axis is from the origin toward the top (towards midpoint of *j0503 and *j0501). The part z-axis is vertical and the y-axis is generated from the right hand rule.

The location of these points on the jacket model can be seen in Figure 11. The commands used to generate this view are shown below the figure:

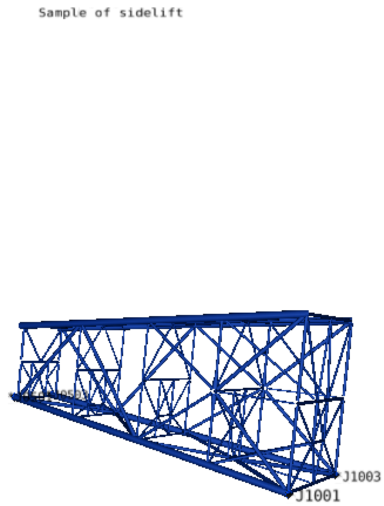


Figure 45: Iso view of points used to redefine the part coordinate system

```
&select :ppp -select *j0501 *j1001 *j0503 *j1003
&picture -90 -25 0 -render gl -water no \
                  -body jacket -points :ppp -anotate points
```

After reorienting the jacket, the next set of commands locates the bodies relative to each other. When locating the jacket, keep in mind that we are now using the new jacket origin and coordinate system. The *&instate* command is written on two lines, however with the line continuation character "\," it is all one command. The barge is set at a 20 foot draft, but its bow centerline is kept at the global X-Y origin. The jacket origin is set 143 ft in the positive global x-axis (143 ft aft of the barge bow); 200 ft on the positive global y- axis (200 ft starboard of the barge centerline); and, finally, the origin (the midpoint of the vector between nodes *j1003 and *j1001) will be 1 foot above the water level with a –9 degree pitch (the top elevation will be pointing up).

Now that the bodies are positioned, we can define the connectors. The connectors are added in the model editing menu which is entered with the command *medit*.

First, we define the classes. We will need classes for the boom line, the slings, and the hold–back lines (tuggers). We define the classes ˜boom and ˜sling as having an outer diameter of 3 inches and a length of 200 ft; and the class ˜*airt* to simulate the hold–back lines. In this case, we will simulate the hold–back lines with a small tug boat, but we will refer to it as an air tugger for this analysis.

In the next set of commands, two air tugger connectors are defined as connecting to the jacket top at point *j0501. The values used in the *–tug* option indicate direction

and length. Essentially the combination of tugs are going to act in the 45 degree direction.

The next set of commands define the tip–hook assembly. The tip-hook assembly consists of a boom line and four slings. First, we attach one end of the boom line to the boom tip and one end of each sling to the appropriate point on the jacket. This is what is done with the five *connector* commands defining boom: sling1, sling2, sling3, and sling4. Now, attach all 5 segments to a common hook. This is much like attaching the hook hanging from the boom to the four slings attached to the jacket. The attachment of the 5 segments is done with the *assembly t–h_definition* command.

At the end of the *assembly t-h_definition* command, there is the option *-initial*. From the manual, we learn that using this option instructs MOSES to move the body so that the hook point is directly below the boom point. So far, we have not checked the geometry to verify the designated sling lengths will work with the position of the bodies. For now, the *–initial* option only places the hook below the boom. We are not expecting the slings to be tensioned or slack.

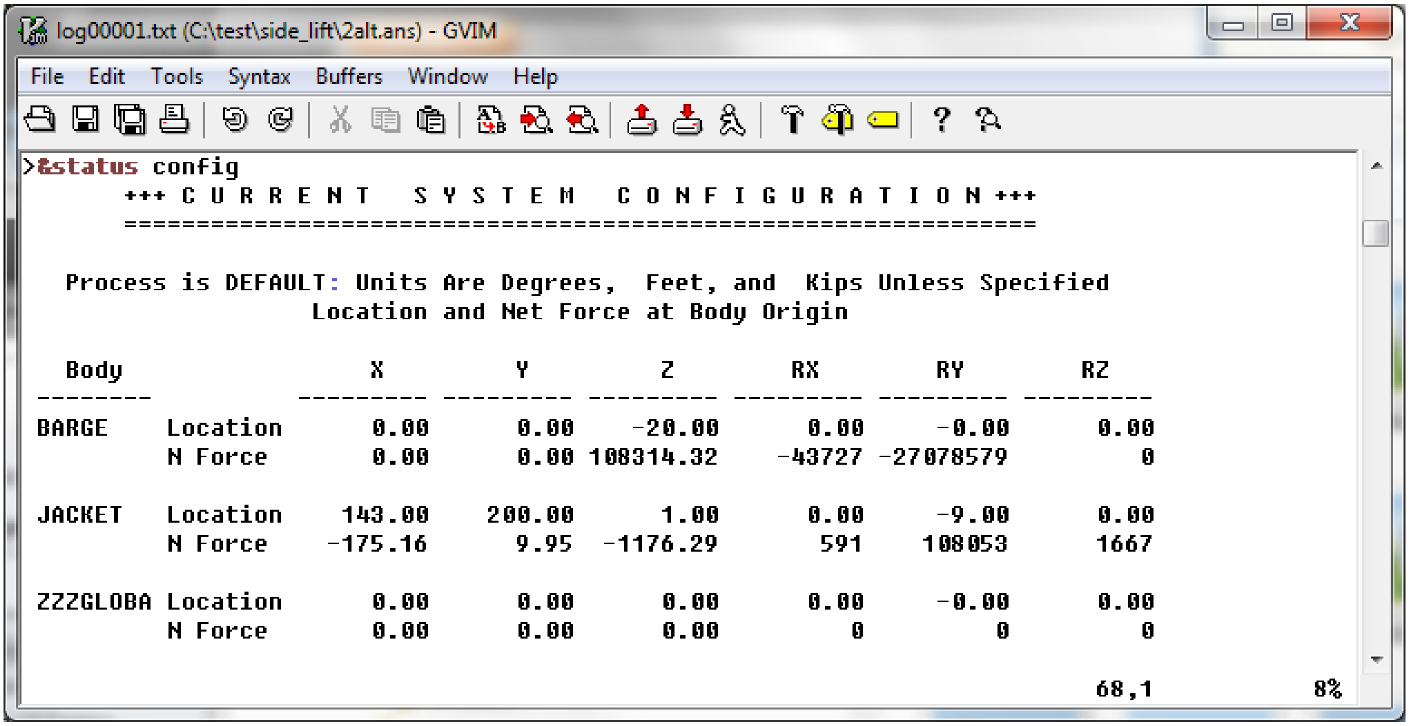
Now that our connectors are defined, we can exit the model edit menu and complete the analysis setup. To exit the model edit menu, we issue the *end* command.

Sidelift Command File Discussion - Connecting the Two Bodies

Now that we are back in the Main Menu, our objective is to set the system in static equilibrium in preparation for the dynamic analyses. The first thing we want to do in the main menu is check the configuration. We do this with a series of *&status* commands:

```
&status config
&status cl_flex
&status g_connector
&status tip-hook
&status b_w
```

The results of these commands are shown in the log file. The config report (Figure 45) will show us the position and any forces in each body. When you first review the resulting table, you will notice that the body ZZZGLOBAXES is listed and that all the values are zero. Since we are interested in the values associated with the barge and jacket, we can see from the results that the resultant forces and moments (N Force) are actually quite large. Our system is not in equilibrium.



log00001.txt (C:\test\side_lift\2alt.ans) - GVIM

File Edit Tools Syntax Buffers Window Help

```
>&status config
     +++ C U R R E N T   S Y S T E M   C O N F I G U R A T I O N +++
     ===============================================================

  Process is DEFAULT: Units Are Degrees,  Feet, and  Kips Unless Specified
                      Location and Net Force at Body Origin

  Body                    X         Y         Z        RX        RY        RZ
--------              --------- --------- --------- --------- --------- ---------
BARGE    Location        0.00      0.00    -20.00      0.00     -0.00      0.00
         N Force         0.00      0.00 108314.32    -43727 -27078579         0

JACKET   Location      143.00    200.00      1.00      0.00     -9.00      0.00
         N Force      -175.16      9.95  -1176.29       591    108053      1667

ZZZGLOBA Location        0.00      0.00      0.00      0.00     -0.00      0.00
         N Force         0.00      0.00      0.00         0         0         0

                                                                  68,1          8%
```

Figure 46: Results of *&status config* command

The next three reports (*cl_flex*, *g_connector*, *tip-hook*) show details of the connector classes, connector geometry, and sling assembly geometry, respectively.

It is also a good idea to review the buoyancy and weight report. This is what is presented with the results of the *&status b_w* command. This report shows the weights, buoyancies, and their centers. From the results of *&status b_w*, we learn that the jacket weight is 1190 kips.

We are going to continue the search for static equilibrium and we will continue to produce these tables until we are satisfied that all of the results can be explained.

In the next command, *&connector &boom –l_tension 1000* we change the boom length until the tension is 1000 kips. This is the maximum capacity of the crane. Right after changing the boom line length, we ask for a report of the forces in the connectors *&status f_connector*. This shows us that the force in the boom sling is 1000 kips.

However, the vertical force in the slings sums to 988 kips (310.7+310.7+183.3+183.3). Also, the *&status configuration* table shows the jacket to be 188 kips out of equilibrium, which we know the slings were not holding.

So, what's up? Notice that the forces are reported in the body coordinate system. Since the jacket has a 9 degree pitch, Z_body is NOT parallel to Z_global. If we do the math, the force of the jacket in the global z direction is 1190.63 * cos(9deg) = 1176.02 kips. This is where the 188 kips net force comes from.

With the command *&status tip-hook*, we see what change was done to the boom length. Originally, the boom length was defined as 200 ft. We see from the report

that the length has now been changed to 221 ft. This change resulted from the *&connector &boom –l_tension* command.

Now that the slings and boom line are set up, we need to make sure the barge is in equilibrium. If we had compartments modeled, we could change our ballast configuration. However, since we don't, we can add a weight. In the next command, *&weight –compute* we change the mass properties of the barge so that it is in equilibrium in the current state. The current state includes the lightship weight and CG location, the buoyancy force, and the force from the boom. After this command, the report from *&status config* shows that the body barge is in equilibrium.

To verify that the sling assembly has not changed, we again report the forces in the connectors with *&status f_connector*. We also ask for a report of the position of joint *J1001 to make sure the jacket is out of the water.

At this point, we know that the body barge is in equilibrium but the body jacket is not. We are going to let MOSES change the location (translation, and rotation) in our next attempt at finding equilibrium. Before we do that, we need to turn off the airtuggers. We do this with two command lines in the command file. It could be done with one. This way we get to see how to make selectors with the wild character (@).

First, we make a selector *:air* with the command *&select :air –select airt@*. This will create a set of items that begin with airt. The command *&connector :air –inactive* turns off (*inactivate*) the airtuggers. This way, when we ask MOSES to find equilibrium for the jacket, only the sling forces, the weight, and buoyancy of the jacket are used.

Previously, we ensured that the barge is in equilibrium. Now, we want to exclude the barge from any changes when the jacket is being altered for equilibrium. The *&describe body barge –ignore x y z rx ry rz* tells MOSES to ignore the barge when calculating equilibrium.

The command *&equi* will change the position (and orientation) of the jacket with the objective to find equilibrium. This command makes 50 attempts at finding equilibrium. If equilibrium is not found within tolerance, it will report a WARNING message. At the conclusion of the equilibrium calculations, the command *&status configuration* produces a report which shows that the jacket body is in equilibrium, but the barge body now is not in equilibrium. We also see that the vertical position of the jacket has changed, moving the jacket into the water. Since part of the jacket is in the water, the tension in the slings changes, which in turn causes a change in tension in the boom line. Therefore, the results of the *&weight –compute* which were based on a boom tension of 1000 kips are no longer valid. Now we see that the barge is out of equilibrium by nearly 188 kips.

Since we are trying to analyze the case where the jacket is in the air, we need to check to verify that the jacket is above the water. We can verify this by checking the location of one of the jacket corners, specifically the location of *J1001. To make

this check, we use the string function *&point(coordinate *j1001 –g)*. String functions actually query the database and return the values asked. We need to ensure that we are telling MOSES what to do with the return values. In this case, the command *&type location of *j1001 = &point(coordinate *j1001 –g)* tells MOSES to put the results in the log file which shows "location of *j1001 = 155.8654 248.6527 -4.199845." We can see from the coordinates of *j1001 that the jacket is not completely out of the water.

In order to analyze the case with the jacket above the water, we are going to again change the length of the boom sling. This is done with the *&connector &boom –l_delta* command. The boom sling will be shortened by 8 feet. This should lift the jacket from the current –4 feet to a +4ft.

First, we check that the boom sling has been changed by 8 ft. Earlier, the command *&status tip-hook* reported 221.93 ft for the boom line length. We see that now after the change it is 213.93 ft. Next, we check the tensions on the slings with *&status f_connector* and find that the sling tensions are very large. This is a result of the boom sling being shortened but the position of the jacket not moving to accommodate the shortened length. The solution is to use *&equi* again so that MOSES can reposition the jacket.

The results of *&equi* show that the jacket has been moved and that the barge is still out of equilibrium by the same 189 kips. Next, we will check the position with the same string function *&point(coordinate *j1001 –g)* that we did before. We see the results now show that the position of joint *j1001 is not 4 feet above the water line, it is 1 foot above the water line. The reason for this discrepancy is that when the jacket was moved (with the *&equi* command), it was moved in all 6 degrees of freedom. In our previous results of *&equi*, the jacket pitch was –11.94 degrees and now it is –13.62 degrees. Now that we have resolved the pitch question, we can conclude that the jacket is above water as desired.

Now we will deal with the barge out of equilibrium issue. For this, we bring the barge back into the calculations. To take the barge out of the calculations, we used the command *&describe body barge –ignore x y z rx ry rz*. To bring the barge back into the calculations we use the command *&describe body barge –ignore*. When the option *-ignore* is used and the space after it is left blank, it turns on all of the degrees of freedom for that body.

The results of the next *&equi* command show that both the barge and the jacket are now in equilibrium. Note that now the barge has a list to starboard (RX = –0.17 deg), and the jacket has a list toward port. The same report is repeated with the *&status config* command. As before, we also review the results from the *&status b_w* command. The results in these reports will not include the force of the connectors (slings). We are NOT expecting the buoyancy force to equal the weight.

And finally, we activate the airtuggers with the command *&connector :air –active*. Once we have activated the airtuggers, we review the forces on the connectors again and find the force in the boom line is 1184 kips. Since our crane capacity is 1000

kips, we are going to need a larger crane barge. However, for the current lesson, we will assume that the crane is capable of handling the loads.

Before ending the static analysis section, we plot some pictures. The three commands that begin with *&picture* will save the views: starboard, bow, and top. Now, we get to talk about the body ZZZGLOBAXES. In the starboard view of the system, you will see thick green arrows which represent the X and Z axes. From the top view, you will see the X and Y axes so that the body ZZZGLOBAXES acts as a visual reminder of the global system.

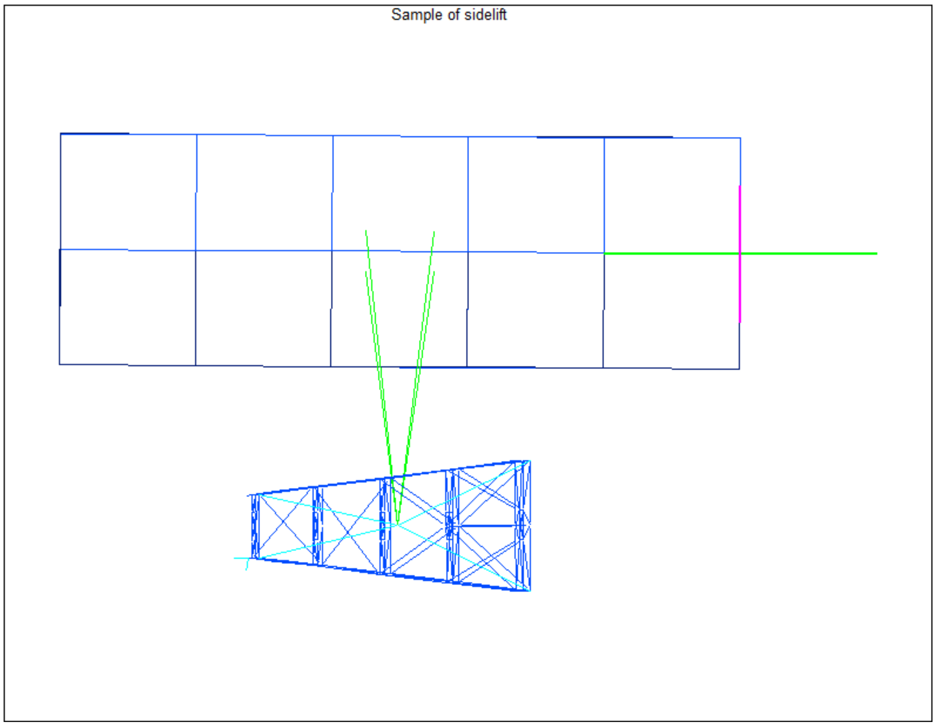


Figure 47: Top view of configuration with global X and Y axes

Having those three views, we finish the static analysis portion of the command file.

Sidelift Command File Discussion - Dynamic Frequency Domain Analysis

The dynamic analysis portion of the command file begins with the command *hydrodynamics*. This command enters the Hydrodynamics Menu where the hydrodynamic database will be computed.

The database is a description of the panel pressures below the water surface. To generate the pressure database we issue the command *g_pressure*. There are many options that can be used with *g_pressure*, but we are using the bare minimum to complete a dynamic analysis. We specify one environment heading and a very small number of periods. If this were a real world project, both of these lists would be longer and the periods would not be evenly spread. Instead, they would have a concentration around the peak period with an overall spread to cover the energy of the expected wave spectrum.

In the log file after the *g_pressure* command, we see a very short report of what MOSES is doing. We see that strip theory was used to make the calculations and that only 90 panels were used. Depending on the shape and size of the vessel, we might be interested in 3d diffraction and increasing the number of panels. Since this is an exercise, we are not going to do a study to determine the best number of panels and the best hydrodynamic theory for our problem.

Now that we have a hydrodynamic database, we exit the hydrodynamic menu using *end*. We can compute motions and forces. First, we will perform a frequency domain analysis. This is also known as a linear analysis. All of the frequency domain analysis is done within the Frequency Response Menu which is entered with the *freq_response* command. After we are finished with the frequency analysis, we will exit the Frequency Response Menu with the command *end*.

You have to be careful about using the command *end*. This command is used to exit many of the menus. Within the Frequency Response Menu, we will enter the Disposition Menu. We need to make sure we keep track of the use of *end* so that we do not accidentally exit into a menu and then have to re-enter the one just exited.

Within the Frequency Response Menu, the **FIRST** thing we do is calculate the response amplitude operators. This is simply done with the command *RAO*. This command computes the RAO but does not report them. MOSES will compute the RAOs at the body origin.

Once the RAOs are computed for each body, we have to tell MOSES which body we are interested in before we ask for the report of the motion RAO. This is why there is an *&describe body bname* before each of the *fr_point* commands. The log file shows that for each body, the RAOs were reported at the local x = 0, y = 0, and z = 0 location (the body origin). Since these are the default values, we are just showing the command order to report the different RAOs. We are not trying to fully scrutinize the response. The command *fr_point* makes any necessary final calculations to translate the motion RAOs to the point specified.

Notice that after each *fr_point* command, you are placed in the Disposition Menu. Within the Disposition Menu, the command *report* generates the standard motions RAO report and puts it in the output file. In the out file, you have two reports titled:

```
"M O T I O N   R E S P O N S E   O P E R A T O R S"
```

The third line of each report reads:

```
"Of Point On Body {body name} At X = 0 Y = 0 Z = 0"
```

where {body name} is either BARGE or JACKET. The reports show the calculated values for the RAO of each degree of freedom. These values include both amplitude and phase. When we review the results of *fr_point*, we see that the peak response of the barge for sway and heave occurs at 11 seconds, while the peak response for roll and pitch occurs at 8 seconds. The peak response of the jacket for sway occurs at 9 seconds, while the peak response for heave and roll occurs at 8 seconds.

The reports of the JACKET RAOs are really response amplitude operators, they are not the results of hydrodynamics on the jacket. The JACKET is being forced via the motion of the tip hook and sling connectors.

For now, that is all we are interested in. Exit the Disposition Menu with the command *end*, which puts us back in the Frequency Response Menu.

We next report the motions of the point designated in the *fr_point* command. We do this with the command *st_point –sea issc 90 10 4 –e_period 5 6 7 8 9* followed by *report motion*. This command calculates the statistics of the body motions for a sea with an ISSC distribution, in the 90 degree heading to the vessel, with a 10 ft significant wave height, and a 4 second mean period. The option *–e_period* instructs MOSES to consider the additional mean periods listed. Again, we are placed in the Disposition Menu where we tell MOSES to report the motion statistics (*report motion*) and exit the Disposition Menu (*end*).

In the out file, we can review the results of *st_point*. Here we have a report titled:

```
"M O T I O N    S T A T I S T I C S"
```

Here the third line of each report reads:

```
"Maximum Responses Based on a Multiplier of 3.720"
```

This tells us on what statistic the maximum values are based. In this case, the maxima have a multiplier of 3.72 which corresponds to $A_{1/1000}$. The multiplier for other values can be readily derived from the derivation of the Rayleigh distribution. They are also shown in the document "HOW MOSES DEALS WITH TECHNICAL ISSUES" which is included in the MOSES distribution (hdesk/documents/include/deals.pdf).

We see that motions for the barge increase as the wave period increases. For the jacket, however, we see the motions are mixed. For sway, the largest motion is 9 seconds. For heave and roll, the largest response is around 6 seconds. What catches our attention is that the maximum dynamic response in sway is 69 ft. If you recall the beam of the barge is 170 ft, the bottom elevation of the jacket is 96 ft. The jacket origin was placed 200 ft starboard of the barge centerline. This leaves 69 ft of clearance. The report we are reading tells us that there is the possibility of collision.

Following the motion reports, we next report the forces in the connectors. The connectors do not belong to a body so it does not matter if the last *&describe body* command was for the jacket or for the barge. First, we get the frequency response for the sling labeled sling1 (*fr_cforce sling1*). Then, we get the frequency response for the boom sling (*fr_cforce boom*).

In the last set of commands, we ask for the statistics of the connector forces in the defined sea state. This is done in the command *st_cforce @ –sea issc 90 10 4 –e_period 5 6 7 8 9*. Like the motions, this calculates the statistics of the connector forces for a sea with ISSC distribution, in the 90 degree heading to the vessel, with a 10 ft

significant wave height. The mean periods of the distribution that will be considered are those listed from 4 to 9.

When we review the results of *fr_cforce*, we see that the peak response occurs at 8 seconds. This is in the out file table titled “Connector Force Response Operators.” This means that a wave with a mean period of 8 seconds will reinforce the response. By reviewing the connector force statistics results of *st_cforce*, we see that the highest force is reported around the 8 second period. These results are in the out file in the report titled:

```
"C O N N E C T O R   F O R C E   S T A T I S T I C S"
```

We want to review the statistics of the forces because the wave properties will change during any operation. It is best to get the reactions to a set of expected waves. This also helps us check if there are concerns with a change in the frequency of the waves during the operation.

Reviewing the manual, you will find that there are several commands that start with *fr_* and *st_*. The *fr_* indicates the frequency response will be calculated. The *st_* indicates the statistics that are based on the frequency response will be calculated.

This concludes the frequency domain section of the command file. Next, we will look at a time domain analysis.

Sidelift Command File Discussion - Dynamic Time Domain Analysis

In the frequency domain analysis, we could define the wave spectrum as part of the statistics command. For the time domain analysis, we must define the environment with the *&env* command. The format used is the same in *&env* and the option *st_cforce –sea*. We see that the spectrum with an 8 second mean period wave is what will be used in the time domain analysis. The option *–time 100 0.2* tells MOSES that the time domain analysis will look at 100 seconds at 0.2 second intervals. The time interval chosen here is usually considered rather large, but since this analysis is just an example, these values will not take long to compute.

The time domain analysis is performed with the command *tdom*. In the log file, you will see the message, “Time To Set Up Convolutions.” Then, MOSES reports when it saves the database and where it is in the event sequence. The last message, “Simulation Terminated at Specified Time,” tells us that the time domain analysis computations are finished. Note that MOSES has performed the calculations for the time domain analysis and stopped. Reporting the results will occur later in the analysis.

In the next three commands, we find when a slam occurred during the first 50 seconds. As an input to the *&slam* string command, we need to know the name(s) of the parts that we want slam information. For our analysis, this just happens to be the one part “jacket.” We made a selector here to show how selectors are used. The next line:

```
&type SLAM 1 to 50 seconds .2 sec increments
```

leaves a note in the log file to remind us of the start time, finish time, and time step. The final line:

```
&type  SLAM = &slam(:lower 1 50  .2)
```

types into the log file the results of the string command *&slam*. This will leave a note and when we review the log file, we see that slam events occurred at times 0, 20.2, 23.2, 28.2, 31.4, 36.8, 41.6, and 47.6. Now the question you are probably asking is, "Why is time 0 in the list?" If we read the manual:

bentley.ultramarine.com/hdesk/ref_man/timdom.htm

we see that the list is a set of pairs where some element of the selected part(s) is submerged between e1 and e2. The result is the list of pairs where the part enters and exits the water. The value 0 is part of the first pair; therefore, it gets reported.

Now we get back to the project requirements; for the structural analysis, we are to use the forces at event 5 and the second occurrence of a slam event. We are going to interpret this as the second event reported by the *&slam* command. We want MOSES to do this automatically for us. During a project, many things can change and we certainly do not want to be running a lengthy time domain analysis then half-post processing the results to get the event at the second slam, and then restarting MOSES to finish post processing. Gathering the list of slam events is considered post processing, however it is done in the Main Menu. For the rest of the post processing of the the domain, we will be doing it in the Post Processing Menu.

This is where we can use variables and the string function *&token*. First, we set the variable f_time to the string of times resulting from the *&slam* command. Then, we use the *&token* command to pick the second value. At the end, the variable f_time will be set to 20.2. Now, we are ready for post processing.

The majority of the post processing will be done inside the "Post Processing Menu" which is entered with the command *prcpost*. What I mean by post processing is somehow getting MOSES to only display the values in the database that are interesting to the project. During the analysis, many items were added to the database. To name a few - (at each time event) wave height, wave force, force on the connectors, position of each body, and velocity of each body. By post processing, we are going to get MOSES to display the values the project wants to examine.

In order to compare the time domain results to the frequency domain results, we need to get the motions and the connector forces. We first get connector forces with the command *conforce*. This command puts us in the Disposition Menu again. Then, we ask for the list of variable names or column headings (*vlist*) that are available. In the log file, we see that there are 57 values available: the events and 8 values for each connector. For now, 57 values are not too much to work with. If we had needed to keep the list of variables to a more manageable size, we could have made a selector to restrict the data to those in which we are interested. The format of the command would be *conforce :sname*.

From the results of *vlist* we that see 1 corresponds to event number, 8 corresponds to magnitude of airtugger 1, and 16 corresponds to magnitude of airtugger 2. The command which makes a plot of the three values is 'textstyleEmphasisplot 1 8 –rax 16 –t_main "Airtuggers" ...' . If you recall, when we defined the airtuggers, we did not give them the ability to change magnitude. So, our plot is going to be two straight horizontal lines.

Next, we plot the magnitude of the boom and the slings. For this plot, we will definitely see some changes as the events change. Note that for both plots, the main title, subtitle, and axes labels are defined. If this is not desired (say you are doing a quick check plot), then this could be omitted and the option *-no_edit* used. The *-no_edit* option tells MOSES to use the default labels.

Once we have created the plots, we ask for the standard reports with the command *report*, then exit the Disposition Menu with *end*. Now we are back in the Post Processing Menu.

The plot of the connector magnitudes shows that the sling tensions (right hand axis) can change from 70 to 640 kips. We can compare the maximum number to the 629 kips reported via the linear analysis. Reviewing the frequency domain report headers, we see that the 629 kips is mean plus maximum. We know from our static analysis that the mean value for sling1 is 339 kips. This means that the dynamic portion is 290 kips (629 - 339 = 290). For a linear analysis, the dynamic portion is added and subtracted, which results in a minimum tension for sling1 of 49 kips (339 – 290 = 49). Just comparing linear and non-linear analyses for the the maximum and minimum sling1 tension values leads us to believe a linear analysis is not too bad.

Next, we compare the motion results.

We do a similar set of commands for the *trajectory* menu. We get the association of the numbers with the column headings with *vlist*. We use this information to get a plot of the barge motions, then we get a plot of the jacket motions. Review of the plots shows us something that we could not see in the linear analysis. The system as defined does not have a mooring system. Therefore, the system wanders in the negative y direction (it is being pushed by the 90 degree waves). The range of motion of the barge is 60 ft and the range of motion of the jacket is 150 ft. We need to look at the phasing of the two motions to see if there is a strong possibility of collision.

To monitor this we will use a variable named "bang." To begin the investigation, we are going to make the assumption that a collision does not occur. So, we set the value of bang to .false.

In the next set of commands, we get relative motion information. This type of data was not available in the frequency response section. The command *rel_motion *ptn1 *pnt2* tells MOSES to find the motion from the first point to the second point. We will be using the three edge points that were designated as points of interest in the data file (*&describe interest –associate *edge/*).

Here it gets interesting because we are using a *&loop* command to cycle through the list. The first time it will use edge = edge1, the second time it will use edge = edge2, and the third time edge = edge3. On all three cycles, the distance to jacket node *j1003 will be measured in the xy-plane of the barge. This means that if the barge has a slight roll, the xy-plane will have that rotation and will not be parallel to the global xy-plane. We will produce a plot of the distance to each edge point separately. Then we will produce a plot of the distances plotted together.

The results of *vlist* in the log file show that for each loop cycle the column (variable) names will only differ by the 1, 2, or 3 after "edge." So, we can use the same column number to get the data of interest. A review of the log file shows that column 5 is always the position magnitude and column 1 is always events. We are using a new command *set_variable*. Up to this point, we have been using the multi–menu form of the command *&set* to define variables. Here the command *set_variable* makes the association with the data that is in the Disposition Menu. First, we are going to find the minimum distance. It does not matter when the minimum distance occurred. The command *set_variable %edge –min 5 5* finds the minimum of column 5 each time the loop cycles. For this discussion, we are going to refer to these as the collision variables. The first time through the loop cycle the collision variable is "cedge1," the second time "cedge2," and the third "cedge3." Individually these variables will be set to whatever the minimum value happens to be for that time through the cycle.

The second set of variables are "redge1," "redge2," and "redge3." We will call these the distance columns. One of these variables is filled each cycle. These variables are populated with the string of values in column 5. We will be using them when we produce the plot of the three distances together.

In getting the values for the "event," we do not need to cycle the name. The list of events is the same regardless of what point is being referenced. It is inefficient to have the same values recorded three separate times. I, however, was not willing to put in the extra keystrokes to make the variable event be filled only once. I left the commands in the sloppy form here.

Back to investigating if the two bodies collide. In setting the collision variables cedge1, cedge2, and cedge3, we were monitoring the distance on the barge xy-plane between the two bodies. If a negative distance is recorded as a minimum, then a collision has occurred. We need a way to ask MOSES if the value recorded for the collision variables is negative. One time is sufficient, we do not care if it occurs multiple times. What we are going to record, or change, is the value of "bang." Once the value of "bang" is changed to .true., we do not want it changed back to .false., and we do not care for it to be reset to .true. a second time.

First, we check the current value of "bang." If this value is **.true.**, then we do not need to change the value and can skip the checking. If the value of "bang" is **.false.**, then we will check to see if it needs to be updated.

Checking is done with the IF statement *&if .not. %bang &then* and ends with the command *&endif*. Within the IF statement, we set the value of "bang" to the results

of the *&logical* statement. The logical statement simply returns either .true. or .false. In this particular instance, we are asking MOSES to see if the value of the collision variable is negative (less than 0). If the value of the collision variable is negative, then "bang" is set to **.true.** and the variable will not be changed again in the loop cycles.

Before the loop cycle completes a plot of the relative position (still working with columns 1 and 5), the distance between the two points is made for each time through the loop. The command *&endloop* tells MOSES where the cycle returns to the top, and for the last entry (edge3) the loop cycle is exited.

We have put a great deal of effort into the collision values. We leave ourselves a note right after the loop to let us know the result. I am referring to the three command lines that begin with *&type*. Review of the log file shows that the values for the variables are substituted. We see from the short message that the barge and jacket did not collide. We also see from our short message that there was a minimum of 8.4 ft clearance between the two bodies. This is actually a big deviation for the 1 ft clearance we determined with the linear analysis. If this were a real project, we would have probably only used the linear response menu to make sure our setup did not cause problems with the software, considered basic checks. The sling elements can go slack, making them non-linear connectors, and therefore we should only consider the results of the non-linear analysis.

In the next set of commands, we are going to be using the *&buildg* menu. This is where we combine the three distance columns into one plot. Recall that we have created the variables redge1, redge2, and redge3 as part of reporting the relative motions. These variables have been populated with the string values representing the distance between the two points at each time step. The main purpose of presenting the gathering of this data in the *&buildg* menu is to show the set of commands to put this new table together. MOSES is a programming language and is intended to be able to analyze many different types of configurations. MOSES comes with preset formatted tables for the analyses considered common. Being able to use the *&buildg* menu is a way to gather data and further process it for the uncommon configurations.

Before entering the Build Graph Menu, we need to know how many rows there are going to be. We will of course let MOSES figure this out for us. Here is another instance where the use of the string command *&token(n string)* comes in handy. The string command determines how many tokens or entries are in the variable event and sets the variable "n" equal to that. We will use *&token* several times in the Build Graph Menu, so it would be worth reviewing the format in the manual.

By this time you have probably figured out that the command *&buildg* will put us in the Build Graph Menu. The command with option *&buildg –brief* here is much like the command with option *plot –no*. We are telling MOSES to just accept our input and not ask for verification. Since we are using the *-brief* option, we are going to have to pay close attention to format. The next four lines with commands (no comment character) are the labels for the column headings.

The next line is blank. It is important that the line immediately after the last column heading is blank (the comment character is after the blank line). This is how we tell MOSES that the list of column headings has ended. As the comment in the command file reads, the next set of commands populates the table. We are using a loop again. We start with the first event and the value for each edge associated with that event. The table is being populated one row at a time. A row is populated each time through the loop. When the data input has been completed (jjj = n), the loop is exited and we have another blank line. The blank line is important here also. This is how we tell MOSES that the data input has finished.

When you review the log file for this section, you will see that this section of the log file is blank. MOSES usually does not echo to the log file inside the *&loop*.

After the loop, we see the results of *vlist* are as we input in the lines above. Finally, we use the plot command to make the plot with all four sets of data on one plot. Then, we exit the Build Graph Menu with *end*.

The next command is just a message to ourselves to make sure we are in the Main Menu. We will be doing a structural analysis to satisfy the final project requirement. The only way to enter the structural solver is through the Main Menu, so we want to make sure that is where we are. Inside the structural solver, we will need to specify which restraints to include in the structural solution. Since we are only looking at the jacket structure, only the four slings attached to the jacket are required. This is why we need to set the selector :restraint to only select the slings attached to the jacket.

We enter the Structural Solver menu with the command *structural*. We tell MOSES which load cases to use with the command *lcase –process*. Remember, we have set the value of f_time to the second occurrence of a slam event, and the project requirement is to have a load case at time event 5. In the next command *s_rest*, we tell MOSES which restraints to use for the structural solution. This is followed by the command *s_part* which tells MOSES on which part to perform the structural solution. If this had been a single body analysis, we would not have had to be so specific with all of these commands. Finally, the commands *reduce* and *expand* perform the structural analysis.

Like many processes in MOSES, we need to first perform the analysis, then ask MOSES to report the results. So, we exit the Structural Solver Menu with the command *end*.

To post-process, we enter the Structural Post-Processer Menu with the command *strpost*. Once inside the Structural Post-Processor, we ask for the results of the beam code check with the command *beam_post code_check* and a summary of the restraint loads with the command *restraint loads*. We see from the WS Beam Check Standard table that the loads created by event 5 dominated for many of the beams.

Exercise A

Perform statistics on the time domain connector force (boom and slings) results. Compare these to the frequency domain results.

Suggested Answer:

```
conforce
   vlist
   statistics 1 24 32 40 48 56 -hard
end
```

The statistics of the boom and slings are reported in the output file.

Remember to compare proper values. The period is 8 seconds. The frequency domain reports the mean + maximum response so that should be compared to the "Maximum" and "Minimum" values from the time domain. The output is shown below:

Frequency Domain

Period	Name	FX	FY	FZ	MX	MY	MZ	MAG.	Ten/Brk
8.00	AIRT1	17.68	-103.97	-35.32	0	0	0	111.22	11.1223
	AIRT2	27.21	13.19	-41.04	0	0	0	50.98	5.0982
	BOOM	64.34	362.43	-1414.90	0	0	0	1461.99	3.4472
	SLING1	-159.49	-272.54	487.46	0	0	0	580.80	1.3694
	SLING2	-125.66	297.68	403.66	0	0	0	517.05	1.2191
	SLING3	359.63	-237.81	344.28	0	0	0	551.74	1.3009
	SLING4	383.80	235.30	379.56	0	0	0	588.84	1.3884

Time Domain

Description	MAG BOOM	MAG SLING1	MAG SLING2	MAG SLING3	MAG SLING4
Mean	1151.32	335.45	340.27	336.33	346.28
Av Of 1/1000 Highest	1770.63	639.58	614.43	581.25	492.39
Av Of 1/1000 Lowest	379.77	73.58	77.51	126.03	150.18

We see that the magnitudes of all connectors are greater in the time domain.

Exercise B

There is also a command *p_min_dist*. We need to make some changes for it to work. In the data file in the barge definition (line 41) change it to read:

```
pgen barge -cs_curr 1 1 1 -cs_wind 1 1 1 -tanaka 0
```

In the data file add to then end

```
&select :jjj -select *j@
```

To the command file add the following to the end.

```
prcpost
  p_min_dist barge :jjj
   vlist
   stat 1 2
  end
end
```

Compare the minimum distance to the one qe calculated earlier just keeping track of 3 points. Did you notice this took the computer a long time to calculate?

Exercise C

Perform relative motions and slamming calculations/plots without loop or the &build_g menu.

Suggested Answer:

```
&set bang = .false.    $  start off assuming it does not bang
rel_motion *edge1 *j1003 *edge2 *j1003 *edge3 *j1003 -mag x y
    vlist
    set_variable cedge1 -min  5  5
    set_variable cedge2 -min 17 17
    set_variable cedge3 -min 29 29
    &set bang1 = &logical(%cedge1 .lt.  0)
    &set bang2 = &logical(%cedge2 .lt.  0)
    &set bang3 = &logical(%cedge3 .lt.  0)
    &if .not.  %bang &then
        &if %(bang1) &then
            &set bang = .true.
        &elseif %(bang2) &then
            &set bang = .true.
        &else   %(bang3) &then
            &set bang = .true.
        &endif
    &endif
$
$ plotting - individual curves
      plot 1  5 -t_sub "Relative Position" \
                -t_x     "Time (sec)"      \
                -t_left  "Distance :  *edge1 *j1003"
      plot 1 17 -t_sub "Relative Position" \
                -t_x     "Time (sec)"      \
                -t_left  "Distance :  *edge2 *j1003"
      plot 1 29 -t_sub "Relative Position" \
                -t_x     "Time (sec)"      \
                -t_left  "Distance :  *edge3 *j1003"
$
$ plotting - combined curves
      plot 1  5 17 29 -t_sub "Relative Position" \
                      -t_x     "Time (sec)"      \
                      -t_left  "Distance to *j1003" \
                      -legend 1 edg1                \
                      -legend 2 edg2                \
                      -legend 3 edg3
end
```

6 Advanced Exercises

This is the advanced section of the workbook. It is assumed that the reader does not need to be given a link to the commands in the MOSES reference manual.

It is assumed that the reader needs a discussion on the command structure. The exercise discussions from here on will focus on why the commands chosen were put in the order they are presented and what project questions were being addressed.

In this section, you will find commands where I refer to my own preferences. I have tried to be careful and always say that "I prefer . . .," or "the only time I . . .," when I am expressing my habits. I am being careful here because these are my preferences and they will not necessarily apply to your situation or your preferences. Please feel free to change any settings I designate as my preference.

6.1 Transportation Analysis

Topics:

- Transportation analysis with native MOSES commands
- Transportation analysis using MOSES tools

Reference files

tow_native.cif, tow_native.dat, tow_brg.dat, tow_jkt.dat, ..\data\env.dat, tow_auto.cif, tow_auto.dat

Discussion

This exercise presents a transportation analysis in two methods. The analysis is done with native commands, then the same analysis is done with the installation tools. The analysis done with the installation tools is considered more complete. The objective in presenting both methods is to show some of the steps that the tools are using, and to show that if you wanted to go"the long way," you could. An effort is made to make the order of the output file reports in the native command results mimic the order of the output file reports in the installation tools output file.

The output is presented in three general sections. The first section presents a summary of the models. The second section presents the motions results. The third section presents the structural analysis results.

For both analyses, we will be looking at the forces and structural solution of a square looking jacket being transported on a rectangular barge. This is a rigid barge analysis. The jacket should be placed 200 ft aft of the barge bow and 5 feet above the barge deck. There will be supports between the barge and the jacket to transfer forces from the jacket to the barge.

The file also contains comments to explain what is being done. The discussion is meant as a complement to those comments. This discussion assumes that the reader has the data, command, log, and output files available. Many of the commands used have been discussed in previous exercises. This discussion assumes the previous exercises have been read.

Model data file tow_brg.dat and jacket.dat

The same barge and jacket data files are used for both analyses. The barge data file conforms to the requirements for being in the MOSES library. To see the requirements for being in the MOSES barge library, please go to the following link:

bentley.ultramarine.com/hdesk/tools/vessels/vessels.htm

This is a rectangular barge, length = 300 ft, width = 90 ft, and height = 20 ft. Only the outer shell was included in the definition. It has five points defined with

the name beginning with a *v. The origin of the barge is at the intersection of the centerline, bow, and keel. The body has a name of "tow_brg." You will see that in tow_brg.dat, line number 71, the command reads *&descirbe body %vname*. In the file tow_brg.dat, you will need to return to line number 38 to see that the variable "vname" has been set to "tow_brg."

The jacket is a space frame that looks like a square. The jacket is made up of 12 members (only 6 of them were named). There are eight points defined. A point used as part of a beam definition becomes a node, and a node used to define more than one beam is a joint. The terms "point," "node," and "joint" will be used interchangeably. For the jacket, all of the nodes\points begin with "*d." There are a total of eight points defined to join 12 members. Only 6 of the members were given names, all starting with "b." Point *db0 was set at the origin x = 0, y = 0, and z = 0. The jacket is part of the body "tow_brg." In jacket.dat, line 6 reads *&describe part jacket*.

Native Commands Method

Model data file native.dat

The file native.dat consists of two commands, *&insert tow_brg.dat* and *&insert jacket.dat*. When we view the two files, we see that the order of files in native.dat is important. For this analysis, there is going to be one body "tow_brg." The body tow_brg consists of two parts names, "tow_brg" and "jacket." The body "tow_brg" has to first be established before any parts can be added. This is why order is important in the file native.dat.

Command file native.cif

Now, we will discuss the command file. As with other files, the model is read with the *inmodel* command. For this *inmodel*, we are using the *–offset* option. This tells MOSES to take out any "extra" steel that is in the computer model that would not be in the real world. That is to say, the tubular beams are defined from node to node. However, when they are welded the steel will have to be offset due to the bracing needing to join the chord at the outer diameter. The distance from the node to the chord outer diameter would be taken out from the computer model with the *–offset* option.

When the models are read in their part coordinate systems are coincidental. What we want is for the jacket to be placed somewhere on the barge deck. After the *inmodel* command, we place the jacket origin within the barge coordinate system. This is done with the command *&describe part jacket -move 200 0 25 0 0 0*. Remember that the joint *db0 was defined at the jacket origin. The *&describe* command will put point *db0 at x = 200, y = 0, z = 25 without any rotations. The jacket part x, y, and z axes are still parallel to the barge body x, y, and z axes.

In the next command, *&instate -locate tow_brg 0 0 -10 000 0 0*, we are placing the barge body (tow_brg body which includes the tow_brg part and jacket part system)

so that the barge coordinate system is negative 10 ft in the *z* direction of the global coordinate system. More simply stated, the barge is being put at a 10 ft draft.

So far, we have just moved the jacket with respect to the barge. We have not established any connections between them. The vertical supports (cans) and the sea-fastenings (tiedowns) are defined in the next set of commands. All of the cans and tiedowns are referred to collectively as part connectors and are defined within the model edit, *medit*, menu.

For both cans and tiedowns, we will be using tubulars of outer diameter 20 inches and 1 in thickness. We have defined two classes "~vert" and "~tiedown" to make labeling easier.

The definition of the vertical support (cans) connectors begins with the command *&describe part can pconnect*. It is important that "pconnect" is included in this command. For the vertical connectors, we have conveniently left points on both parts that line up vertically to connect. Each of the four vertical cans is defined separately with the command *pconnect*. Basically, we are telling MOSES to start with the jacket point and move in the negative *z* direction to find the corresponding point on the barge. Please note that for the can definition, we were explicit about which node on the jacket and which node on the barge to connect.

At this point, we compute a weight (can be positive or negative) needed for the body system to be in equilibrium. We confirm this weight calculation with the *&equi* command. Then, we define this setup as event 1. This will be the setup used for the still water case later. This is meant to represent the project after loadout but before the tiedowns have been welded.

The definition of the sea–fastenings (tiedown) connectors begins with the command *&describe part tiedown pconnect*. Again, it is important that "pconnect" is included in this command. We will be defining four tiedowns at each corner of the jacket. The tiedowns will have a general 45 degree pattern. On the connector definition, we are only specific about which node on the jacket the connector is to attach. For the barge end, the wild character is used. This leaves the connection at the barge end up to MOSES. MOSES is going to place the barge end of the connector at 4 ft in the x direction, 4 ft in the y direction, and 5 ft in the negative z direction. From this location, it will make a rigid connection to the nearest barge node. How MOSES does this is also discussed in the user's manual. Please see the following link:

bentley.ultramarine.com/hdesk/ref_man/p_conn.htm#PCONNECT

This is all of the model editing we will do. Exit the Model Editing menu with the command *end_medit*.

In the next set of commands, we bring the barge body system (tow_brg, jacket, tiedowns, and cans) into equilibrium. The command *&weight –compute* instructs MOSES to add a weight (positive or negative) such that the sum of the forces and

moments is within the default tolerance. The weight will be added at 10 ft height from the keel, and the x and y location of the weight will be determined by the moment needed. After this command, we confirm the system is in equilibrium with the *&equi* command.

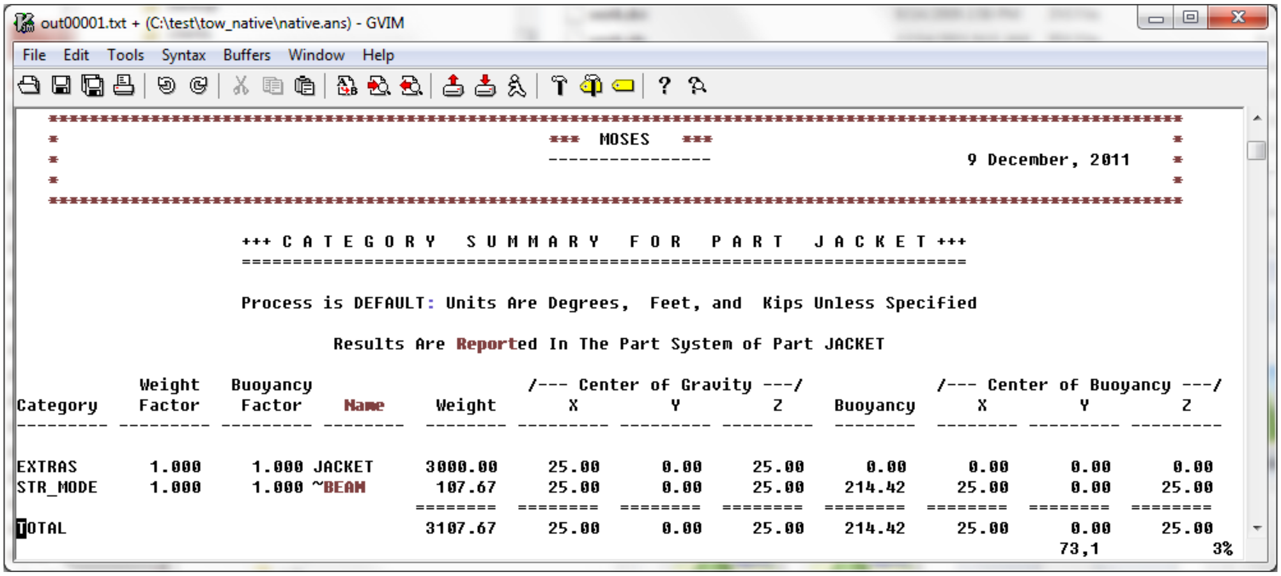
The summary of the input consists of pages 1 through 11. The tables presented in the output file pages 2 through 9 are a result of the commands within the Summary Menu. We enter this menu with the command *&summary*, and exit it with the command *end*.

In this next section of commands, we write several summary reports to the output file. The objective we are trying to fill with these reports is to provide verification to the project that the correct model was used and that we are providing them to parallel the output from the installation tools. Using the command *&rep_sel -part partname* allows us to present the data that is relevant just for the part we are interested in for verification. This makes the output easier to read and hopefully reduces the questions from people reading our results.

For the barge, all that we report is a Piece Summary. This report provides a very short summary on the vessel particulars. For the jacket, we report Category Summary, Wind Vortex Shedding, and Beam SCFs.

For the transportation analysis, there is one body and two parts. If you review the output page 3, and shown below in Figure 5, with title **CATEGORY SUMMARY FOR PART JACKET** produced with the command *compart_sum piece* you should also see that the values are reported in the part coordinate system. If you remember, the barge part coordinate system has its origin at the intersection of the bow, centerline, and keel. The jacket has the part coordinate system origin at point *db0, which right now is located at barge body location x = 200, y =0, z = 25. If you look again at the table, you will notice the column headings indicate where the center of gravity and the center of buoyancy are for each item. Since the part coordinate system for the two parts is different, then reporting them on the same table and then, for the last row, summing the results would be misleading. This is why we want to use the

&rep_sel command for these reports.



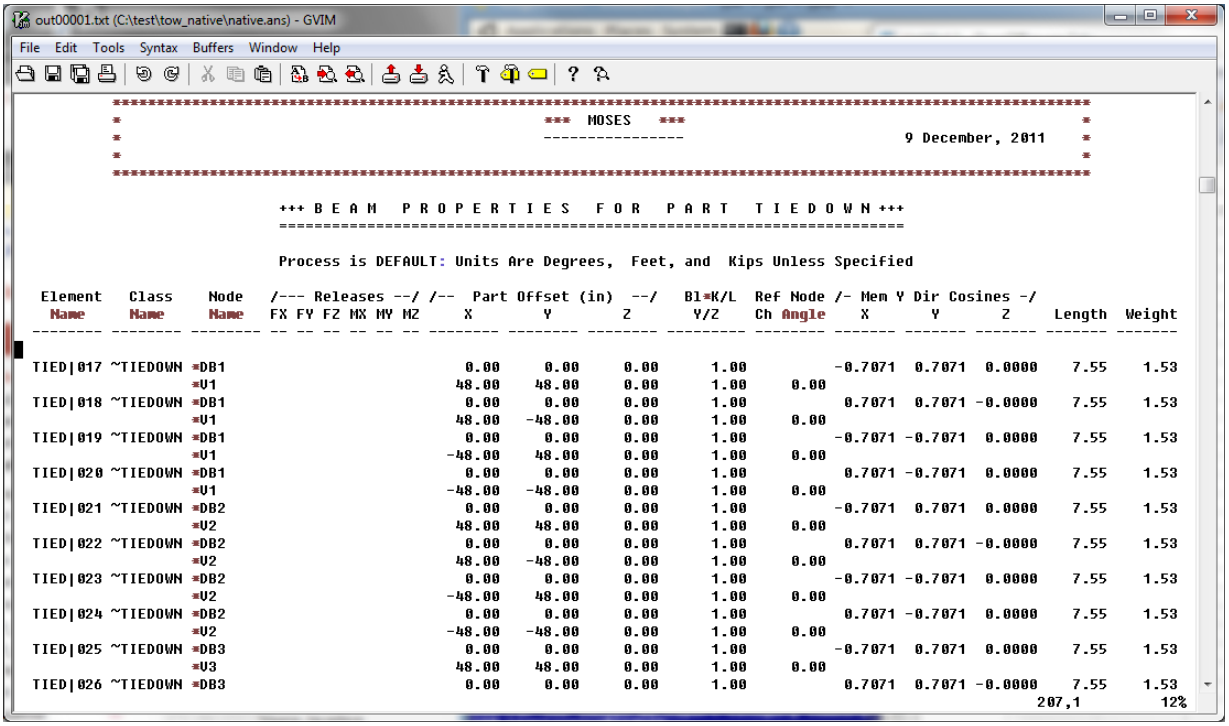
```
out00001.txt + (C:\test\tow_native\native.ans) - GVIM
File  Edit  Tools  Syntax  Buffers  Window  Help

  *************************************************************************************************
  *                                     ***  MOSES   ***                                          *
  *                                     ----------------                  9 December, 2011        *
  *                                                                                               *
  *************************************************************************************************

                  +++ C A T E G O R Y   S U M M A R Y   F O R   P A R T   J A C K E T +++
                  ======================================================================

                  Process is DEFAULT: Units Are Degrees,  Feet, and  Kips Unless Specified

                          Results Are Reported In The Part System of Part JACKET

           Weight    Buoyancy                        /--- Center of Gravity ---/             /--- Center of Buoyancy ---/
Category   Factor    Factor     Name      Weight        X         Y         Z      Buoyancy     X         Y         Z
--------- --------- --------- --------  --------- --------- --------- --------- --------- --------- --------- ---------

EXTRAS      1.000      1.000 JACKET    3000.00     25.00      0.00     25.00      0.00      0.00      0.00      0.00
STR_MODE    1.000      1.000 ~BEAM      107.67     25.00      0.00     25.00    214.42     25.00      0.00     25.00
                                      ========  ========  ========  ========  ========  ========  ========  ========
TOTAL                                  3107.67     25.00      0.00     25.00    214.42     25.00      0.00     25.00
                                                                                                     73,1          3%
```

Figure 47: Category Summary Table

The next report is on classes. Classes do not belong to a part, so the reports titled Class Dimensions and Material Redesign Properties show classes and material properties for the jacket, the tiedowns, and the cans.

For the tiedowns, we have reports titled Beam Properties and Beam Ends. Both of these reports are useful in verifying that the tiedowns are modeled as the project desires. The beam properties summary is where we can verify that the tiedown orientation is as input through the use of the member y-direction cosine matrix. For the beam ends report, we are given the location of each end of the tiedown in the barge part coordinate system. Please note that we are paying special attention to the part system designation for these reports. We want to be careful and provide

proper documentation.



```
out00001.txt (C:\test\tow_native\native.ans) - GVIM
File  Edit  Tools  Syntax  Buffers  Window  Help

   *******************************************************************************************
   *                                  ***  MOSES   ***                                       *
   *                                  ----------------                  9 December, 2011     *
   *                                                                                         *
   *******************************************************************************************

                   +++ B E A M   P R O P E R T I E S   F O R   P A R T   T I E D O W N +++
                   ======================================================================

                   Process is DEFAULT: Units Are Degrees,  Feet, and  Kips Unless Specified

 Element   Class    Node   /--- Releases --/ /--  Part Offset (in)  --/   Bl*K/L  Ref Node /- Mem Y Dir Cosines -/
  Name     Name     Name   FX FY FZ MX MY MZ    X        Y        Z        Y/Z    Ch Angle    X       Y       Z     Length  Weight
-------- -------- -------- -- -- -- -- -- -- -------- -------- -------- --------- -------- ------- ------- ------- ------- -------

TIED|017 ~TIEDOWN *DB1                           0.00     0.00     0.00      1.00          -0.7071  0.7071  0.0000    7.55    1.53
                  *V1                           48.00    48.00     0.00      1.00     0.00
TIED|018 ~TIEDOWN *DB1                           0.00     0.00     0.00      1.00           0.7071  0.7071 -0.0000    7.55    1.53
                  *V1                           48.00   -48.00     0.00      1.00     0.00
TIED|019 ~TIEDOWN *DB1                           0.00     0.00     0.00      1.00          -0.7071 -0.7071  0.0000    7.55    1.53
                  *V1                          -48.00    48.00     0.00      1.00     0.00
TIED|020 ~TIEDOWN *DB1                           0.00     0.00     0.00      1.00           0.7071 -0.7071  0.0000    7.55    1.53
                  *V1                          -48.00   -48.00     0.00      1.00     0.00
TIED|021 ~TIEDOWN *DB2                           0.00     0.00     0.00      1.00          -0.7071  0.7071  0.0000    7.55    1.53
                  *V2                           48.00    48.00     0.00      1.00     0.00
TIED|022 ~TIEDOWN *DB2                           0.00     0.00     0.00      1.00           0.7071  0.7071 -0.0000    7.55    1.53
                  *V2                           48.00   -48.00     0.00      1.00     0.00
TIED|023 ~TIEDOWN *DB2                           0.00     0.00     0.00      1.00          -0.7071 -0.7071  0.0000    7.55    1.53
                  *V2                          -48.00    48.00     0.00      1.00     0.00
TIED|024 ~TIEDOWN *DB2                           0.00     0.00     0.00      1.00           0.7071 -0.7071  0.0000    7.55    1.53
                  *V2                          -48.00   -48.00     0.00      1.00     0.00
TIED|025 ~TIEDOWN *DB3                           0.00     0.00     0.00      1.00          -0.7071  0.7071  0.0000    7.55    1.53
                  *V3                           48.00    48.00     0.00      1.00     0.00
TIED|026 ~TIEDOWN *DB3                           0.00     0.00     0.00      1.00           0.7071  0.7071 -0.0000    7.55    1.53
                                                                                                                207,1        12%
```

Figure 48: Tiedown Properties Summary Table

Pages 10 and 11 of the output are a summary of the SN curves that will be used for fatigue. These are included as part of the first section summarizing the input. For our analysis, we are not changing or adding to this list, but we still need to provide the curves that are used as part of the output. Finally, we make four pictures showing the configuration as part of the input summary. The four pictures are created with the *&picture command*.

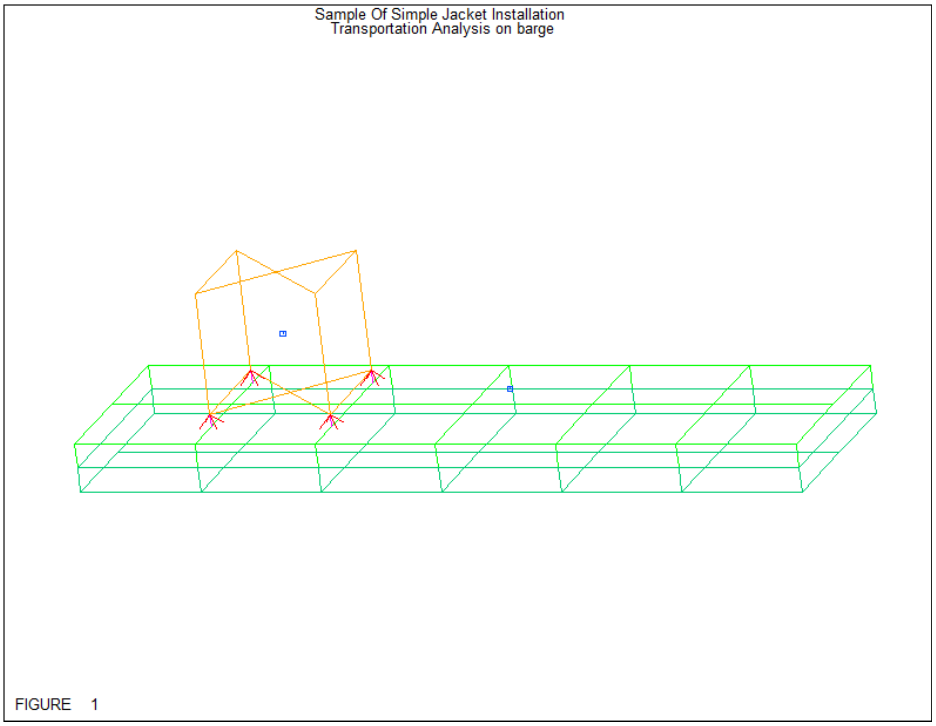


Figure 49: Iso view of Transportation

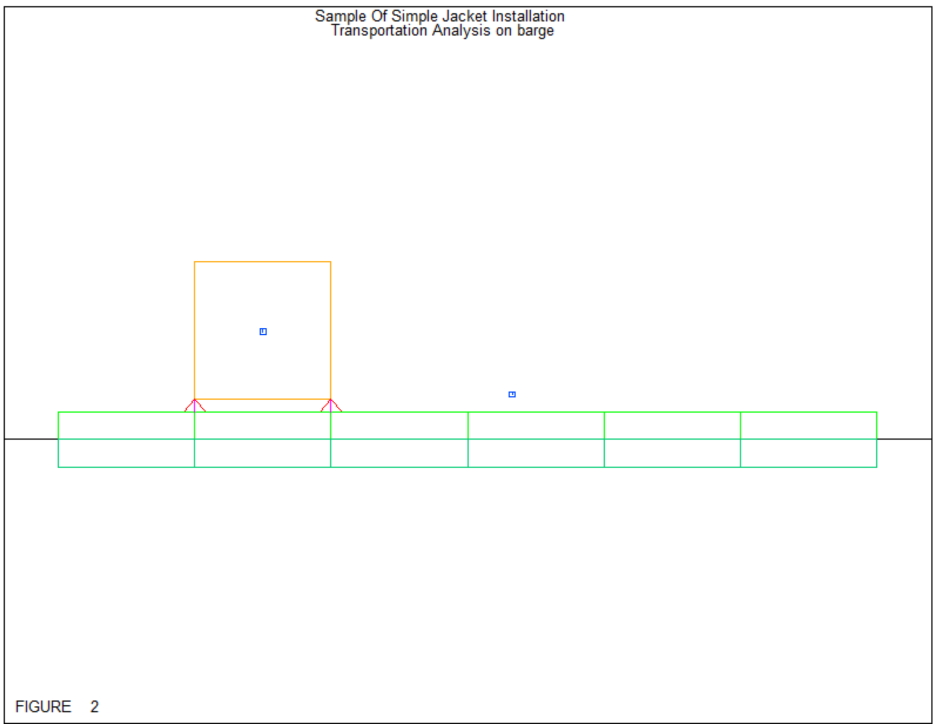


Figure 50: Side view of Transportation

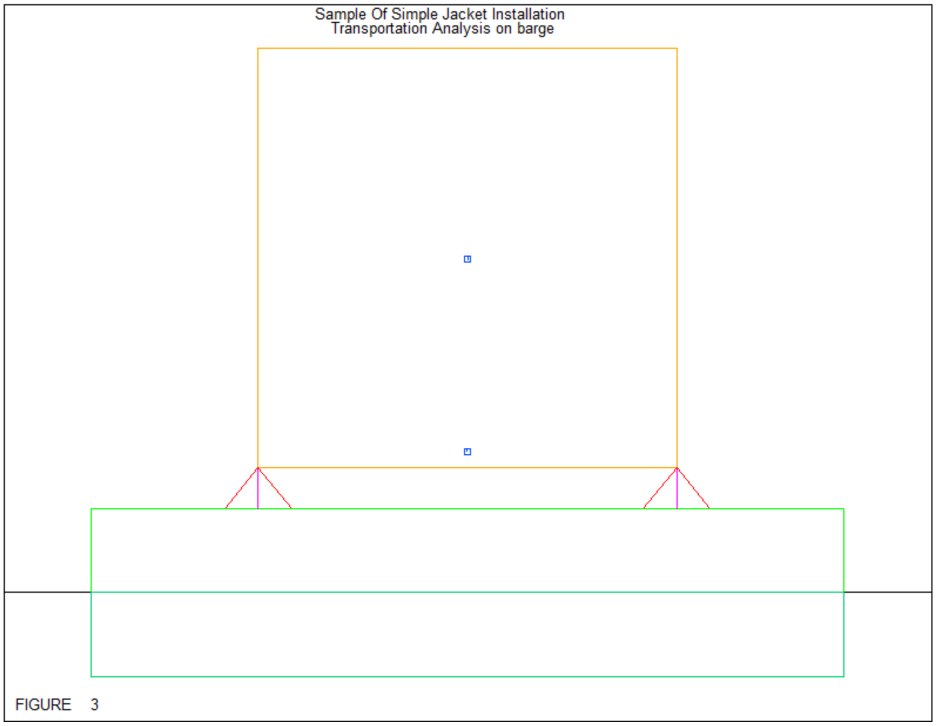


Figure 51: Bow view of Transportation

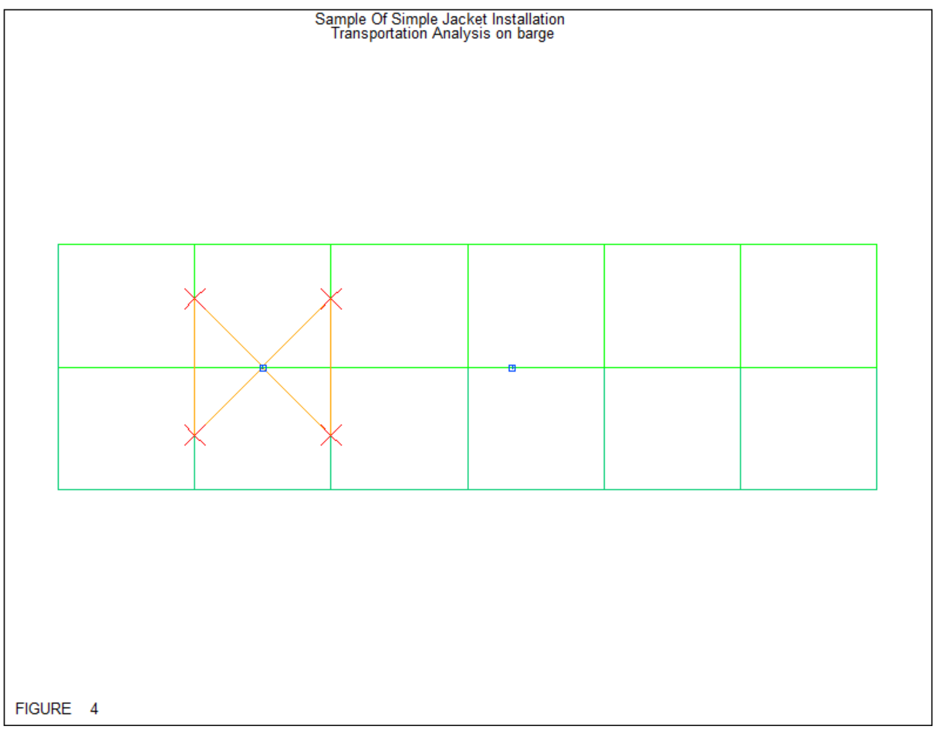


Figure 52: Top view of Transportation

Next, we perform a stability analysis. These are the same commands that were used in previous exercises and will not be discussed here.

In the next section of commands, we produce a set of tables that summarize the system. Please note that all of the values are reported in the barge coordinate

system. The last Category Summary reports in the barge part system. For this report, the structural model has the XCG at 225 ft from the bow. This makes sense because the jacket is located at x = 200 and the XCG in the jacket part system is at x = 25 ft. The structural weight that is reported in the Category Summary report on page 17 includes the jacket, tiedowns, and cans. Also, please note that the command *&weight* added a negative value to achieve equilibrium. The weight added by *&weight* is presented as &DEFWT.

That concludes the static analysis portion of the transportation analysis. Next, we do the hydrodynamics and the structural analysis.

Native Command Hydrodynamic Analysis

Creating the hydrodynamic database has been covered in previous exercises. Basically, the same set of commands are presented here. Please refer to the previous exercises for a discussion on the commands.

The environments we are going to use need to be given a name. The easiest way to do this is in the *&data environment* menu. We are using the same naming convention as that in the installation macros. The first character is a letter designating a Hs and Tm pair, followed by three characters designating the environment heading. Notice that the syntax used for environments in the *&data environment* menu is the same syntax used for the *&env* command. We are going to be using the same environment descriptions for the motions analysis and the structural analysis. Defining them by giving them names, reduces any human error because they are only defined once, it provides a clean way to write the commands, and they can be referenced in the output.

After this, we enter the frequency response menu and calculate RAOs and report RAOs at the system CG. When we calculate the RAOs at the jacket part CG, we do not ask for a report. Then, we calculate the force response operators for the cargo (which happens to be the jacket) and again we do not output the results. This is back to the idea that we are trying to produce the same set of output as the installation tools do.

The next part where we report the G Force Statistics is where we do write reports. The reporting of the G Force statistics is done in numerical and in graphical form. In the output file, the statistics are listed just for the nine environments defined earlier. In graphical form, the period of each of the environments is presented from 4 to 18 seconds. This second part is done with the *–e_period option*.

This concludes the frequency response portion of the analysis. Next, we perform a structural analysis.

Native Command Structural Solution

Before we begin the structural analysis, we reset the active environment to "none" and we take out the subtitle. This is done to keep the output easy to read.

The structural analysis presented in the native command file is the basic approach to a transportation structural analysis. The installation macros contain a great deal of logic that better represents the project progression. The default structural analysis performed with the MOSES tools takes out the tiedowns for the still water case, makes any mean load cases to include the mean wind force, and combines these cases with the dynamic portion.

For the analysis in the tow_native files, a much more limited set of load cases is presented. The structural solver menu is used two separate times. The first time for the still water case, and the second time for the wave environment load cases. We use the same procedure for both instances:

- define the load case with *lcase*
- define the restraints to be used with *s_rest*, will leave blank
- define the parts with *s_part* to be included, this changes
- finally solve for the structural solution with *ssolve*

For both times through the structural solver, we leave the list of restraints to use blank. For this analysis, we are using a rigid barge. The restraints needed to keep the jacket on the barge deck plane are provided by the fact that the barge is rigid. We do not need to include any other restraints since we are only performing a structural analysis on the jacket, tiedowns, and cans.

We define the "still water" case the first time through. This is meant to represent the stage after loadout has finished but tiedowns have not been welded. To do this, the list after *s_part* contains only "jacket" and "can" for this load case. The option *–nonlinear* is used. Many times the connection between the jacket at the cans or tiedowns is not welded. There is a possibility that there could be a temporary disconnect at these locations. This disconnect would make the problem non-linear.

The second time through the structural solver, we define the RAO load cases. This just creates the real and the imaginary cases for a regular wave and the headings. This does not combine the RAO and the wave spectra. This time through the structural solver *s_part* has "jacket," "can," and "tiedown." This is because the dynamic loadcases are supposed to represent a situation where the tiedowns have been added.

At the conclusion of the computation, we just know that we have done the computations. We will have to go into another menu to get a report of the results. That is what is done with the Structural Post Processing, *strpost*, menu. The first thing done in the post processing menu is to combine the RAO load cases with the wave spectra, *cases –spect*. When we review the tables in the output file, the load cases

will have the names listed after the *–spect* option. This concludes the computations for the structural strength analysis. The rest of the commands control the contents of the output file.

First, we make a selector to include all of the spectral loadcases. Then, we report the multipliers for the currently defined structural load cases. This is done with *&status r_case*. The first time through, we report all of the available load cases. This first list is rather long because it includes all of the cases created with the *lcase –rao* command. The second time through, we report only the load cases we are interested in.

The results of the structural code check are presented in various forms in the last set of commands. Usually a project wants to review the static stillwater case separately from the dynamic load cases. Here, we present the results of the code check with the command *beam code –load stillw*. In the next command *beam code –load :load*, we get the summary of the results for all of the load cases. Only the case which resulted in the highest code checks is reported. For our analysis, there is predominantly H090, but there are some H135 still listed. Just to show that the other load cases are available, the last command *beam code –load s135* presents the results of the code check for only the S135 loadcase.

This concludes the strength check of the transportation. The last set of commands computes the fatigue check.

To do fatigue, you need to give MOSES the environments and the duration of each environment. For our tow, we have all of the information in a separate file named env.dat (located ..\data\env.dat0). This file and its format have basically been unchanged for a decade. You need to change the numbers to set the velocity (vel), the total time (tim), and the length (len) variables. After that, you define the environments with the first value being the length of time the transportation is exposed to the environment. Usually the meteorologist will have this information. You will need to somehow put it into this format.

Remember, this file is inserted while we are in the structural post-processing menu. When we end out of the *duration* menu with *end_duration* we can just create the fatigue load case with the command *cases*. Then, when we return to the native.cif file, we just ask for the fatigue report with *beam fatigue*.

After this, we exit the structural post-processing menu with the command *end*. The last thing we do is create a picture of the structural solution. The command *&picture iso –type struct –color ratio* creates a picture with the color of each member of the jacket indicating its value in the structural analysis unity check.

This concludes the transportation analysis with all of the MOSES native commands. Next, we are going to use the transportation tools to do the exact same analysis.

Automated Installation Tools

If you have not read the online documentation for the installation macros, please see the following link:

http://bentley.ultramarine.com/hdesk/ref_man/install.htm

Our discussion will start with the files from the download site, install.dat and install.cif. We are going to start with the install.dat file.

Command file install.dat

The top part of the install.dat file sets many variables which are self-explanatory, wdepth is for water depth, the SCFs to use are from Efthymiou, etc. The variable we are interested in is "envdat" on line 51. This is the variable that tells MOSES where the environment duration for fatigue is located. We want to use the same file "..\data\env.dat" which was used in the native command analysis. We see that the value is already set to what we want, so we proceed.

The changes we are interested in making begin at line 66. We tell MOSES to use the vessel in the current directory in the file tow_brg.dat. Remember, tow_brg.dat was created to conform to the vessel library format. Therefore, this is all we need to tell MOSES when we use the tools.

The next two variables we set are the jacket starboard and the port nodes. If we review the jacket.dat file, we see that nodes *db4 and *db2 have positive values in the y coordinate. When we located the jacket on the barge with the native commands, we did not rotate the jacket in any manner, so the nodes on the starboard side would have a positive y coordinate. This means that nodes *db4 and *db2 were placed on the starboard side. The order that the nodes are placed in is important for these two variables. The node listed first is conventionally known as the leading edge. If we were to imagine that this jacket was to be launched, the nodes at the barge stern would be at the leading edge into the water. We see that nodes *db3 and *db4 have the largest x coordinate and would have been placed nearest the stern in the native command files. Therefore, in our install.dat tool file, nodes *db3 and *db4 are listed first for the variables. To define the port and starboard nodes we have:

```
&set port_nod = *db3 *db1
&set stbd_nod = *db4 *db2
```

This is just the first part in defining the jacket location and sea fastening. The command *model_in* is next used to locate the jacket on the barge. Remember that in the native commands, we also placed the jacket coordinate system origin at 200 feet in the x direction. The z coordinate is what is different. Part of the barge library format is to input a variable "vdepth," which is the distance from the barge deck from the keel. Therefore, the distance used to locate the jacket in the z direction is taken from the barge deck. In the native command file, we located the jacket 25 ft from the keel, but for the transportation tools we will use 5 ft from the deck. The two options used for the command *model_in* use the port and starboard nodes we just designated. *–port_nod* and *–stbd_nod* have the syntax of options, however for

this command they are necessary, they are not options. Using the options *–port_nod* and *–stbd_nod*, we tell MOSES to place nodes *db3 and *db4 nearest the stern of the barge, and place nodes *db4 and *db2 on the starboard side, and nodes *db3 and *db1 on the port side of the barge. For now, that is all we need for locating the jacket on the barge. Next, we will define the supports and sea fastenings.

The next set of commands looks very similar to those that we used in the native.cif file. To define the support cans and the seafastening tiedowns we will be using the *i_connector* command several times. Within the transportation tools, the *i_connector* command and the designators used after it tell MOSES what part is being defined, the orientation, and the connection points. The support cans are defined with the *i_connector v_can* command. Please note that for the transportation tools, we were able to list all of the nodes that will be supported on one command line. This is different from the syntax used in the native.cif file. The classes are defined with the same command as in the native.cif file. MOSES knows to define a part with a "CAN" name when a connector is defined using the designator *v_can*. The tiedowns are defined with the lines that contain the *pconnect* command. These are basically the same commands from the native.cif file with the command *i_connector* before the *pconnect*. MOSES knows that when the command *i_connector pconnect* is used that a tiedown part is being defined.

This is all of the information that is needed to produce the same model as the native.cif file. In the default install.dat file that is located in the download site, there are still several commands after the tiedown definition. Since they are not needed for the comparison, we will not be discussing them here.

Command file install.cif

There are few changes needed to the install.cif file to produce the analysis performed with native.cif. Remember, this discussion assumes that you are starting with the install.cif file from the download site.

The first set of changes are in lines 12 to 16. We are only interested in the transportation analysis. This should be the only variable left with a value of ".true." Please note that the values includes the "." before and after the letters. The values for launch, loadout, upend, and lift should be set to ".false."

The other section we want to change are the options used on lines 33 to 36. In the native file we used wave height and period pairs, 5 and 10, 4 and 11, and 6 and 12. In the native files, we had to specify what headings to look at. In the installation tools, all we have to do is list the Hs and period pair. The installation tools take care of making the environment descriptions for 8 headings (45 degree spacing).

The options *–wind w_inctact w_damage w_vortex w_stuctural* tell MOSES which wind velocity to use for the different parts of the analysis. We tell MOSES to use 100 knot winds for intact stability. We need a value for damage stability. Even though for our analysis we will not be performing damage stability, a 40 knot wind will be used to check vortex shedding and 0 knot winds will be used for structural analysis. The

installation tools check vortex shedding by default. We did not include wind when we defined our environments in the native command analysis. That is why we are also not going to include wind and have used a 0 to indicate this.

The last two options *–draft dd –trim tt* tell MOSES which draft and trim to use. We want to make sure we are using the same values as the native command files, 10 and 0.

That is all of the changes that are needed to the installation tool files. The tools take care of all of the stability, hydrodynamics, and making load cases for us.

Review the answers directory

At this point, we have discussed the approach to the native command method and the installation tools method. Some of the log and output file for the native command method was discussed when the native command file was discussed. We are going to be mostly reviewing the results of the installation tools method with some comparison to the results from the native commands method.

When we review the results in the answers directory, the first thing we notice is that the native commands produced 9 graphic files and the tools produced 22 graphic files. The first five graphics are the same. The first four are shown as Figures 5 to 9 in this workbook. The fifth one is the results of the stability analysis. They are four views of the system and the results of the stability analysis. For the tools results, the RAOs are presented in graphical form in graphic files 6 to 13 and in tabular form pages 25 to 32 of the out file. In the native command results, only the tabular form of the RAO was produced in pages 18 to 20. Remember, in the native command files we only looked at three environment headings, whereas the tools by default will examine eight headings.

The graphs that we can compare are the force response curves. These would be graphics 6 to 9 in the native analysis and some of the figures shown in graphic files 14 to 21 in the installation tools analysis. For the native commands analysis, we did 180 deg (graphic 6), 135 deg (graphic 7), and 90 deg (graphic 9). In the installation tools, they are done in a different order, but they are graphics 18, 17, and 16. You should be able to keep track of them by the graphic titles. When you compare the curves, you will see that so far we are getting the same answers for both analyses types. You can also compare the numerical results in the output file. In the output file, we reported the force response at the jacket CG for just the wave's height and mean period that were specified. These are the tables with the title CARGO G FORCE STATISTICS. When making your comparison, you will need to make sure that the information in the box outlined by *s contains the same information.

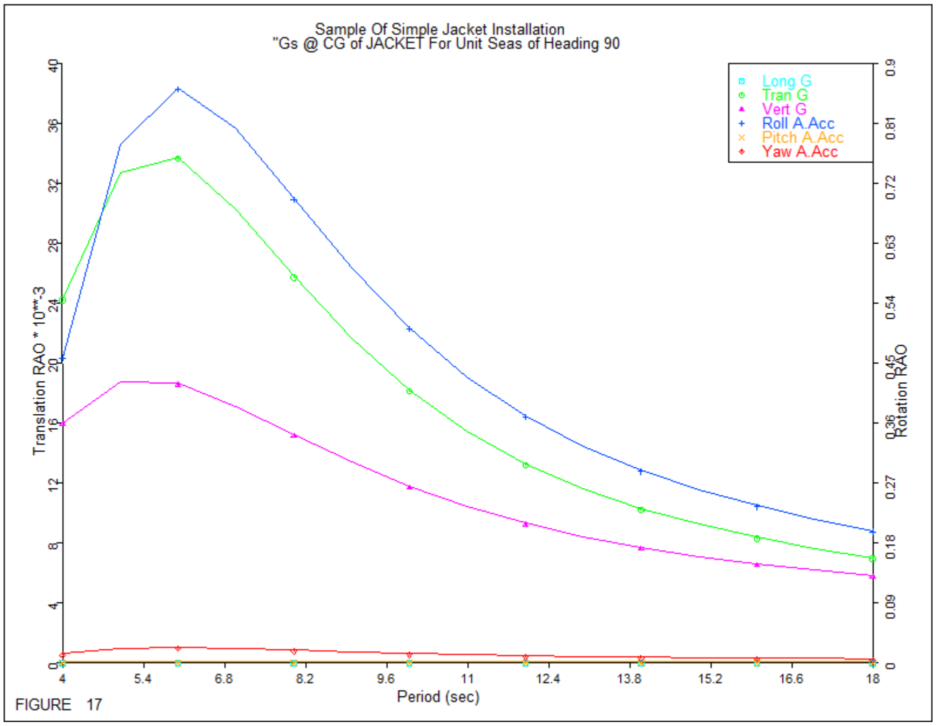


Figure 53: G Force Statistics for 90 deg Using the Tools

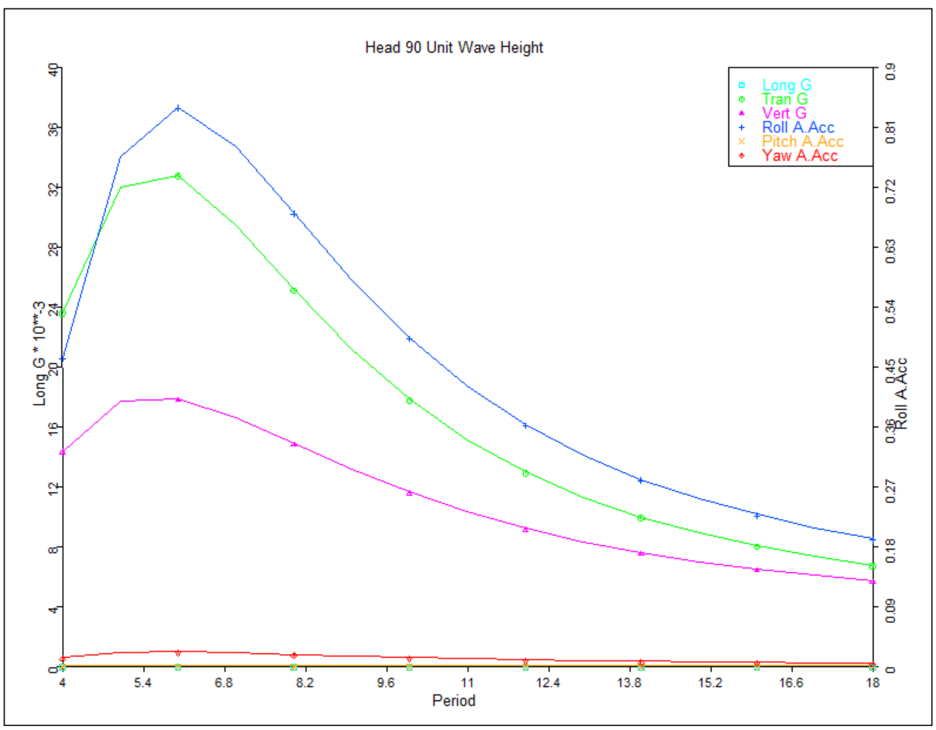


Figure 54: G Force Statistics for 90 deg Using Native Commands

After the hydrodynamics results is where our analyses diverged. In the native commands we made a still water load case, RAO load cases, then we just asked for the default combination of RAOs and environment spectra for the spectral load cases. The report ALLOWABLE STRESS MODIFIERS shows the allowable stress modifiers for the load cases. In the native command method, we did not change it so

the 10 load cases were left with a modifier of 1.00. The corresponding table for the installation tools results is found on page 57. The table shows 24 load cases for the tiedown members with an allowable stress modifier of 1.33. Page 62 shows the loadcases for the dynamic (jacket, can, tiedown system) analysis.

The load case names used in the installation tools is explained in the file doc00001.txt. More sorting and combining is done for the installation tools. If you open the doc00001.txt file, you should be able to find the following explanation of the structural load cases.

> For the structural analysis, the barge was assumed to be rigid, and the behavior of the structure was considered under the action of 282 basic load cases. The structure will be checked for the environments:
>
> | Case | Mean Period, | Sec. |
> |---|---|---|
> | H | 5.0 | 10.00 |
> | S | 4.0 | 11.00 |
> | V | 6.0 | 12.00 |
>
> Each of the above environments are used with the RAO load cases to produce spectral cases. These were obtained by first integrating the product of a member load RAO squared times the spectrum, then multiplying by a probability factor to obtain the dynamic stress. Here, the probability of the average of the 1/1000th highest values was used to compute the dynamic stress. These dynamic stresses are combined with the static stresses to get two load cases for each spectrum and direction. These were named LXXXAS and LXXXAC. Here, the naming convention is that the "L" in the name is the letter corresponding to the spectrum defined above, "XXX" is the direction and "A" is a process designator. The cases ending with "S" are the "normal" ones; they are the wind cases plus the dynamic deviation times the sign of the mean. This governs most members. In some cases, however, the mean of the member may be slightly in tension and the compression cases will govern. Thus, the "C" cases are the "S" cases minus twice the dynamic deviation.
>
> To check uplift, an additional spectral condition was used, LXXXAU. These are the mean plus the dynamic deviation.
>
> A sequential structural solution was performed. First, the system without tiedowns was solved for the still water case (loadout). Then, the system with tiedowns included was solved for the other load cases. The still water load case was combined with the other cases in the post processor so that the effect of the tiedowns is felt only under dynamic loads.
>
> A separate section of the program output is dedicated to the tiedowns.
>
> Here, beam internal loads and code checks for only the tiedowns are shown. The beam loads have been condensed into an envelope. In other

words, each value presented is the maximum overall load case. The load cases used for checking the tiedowns are simply the dynamic loads multiplied by two. This assumes that no tension connection is developed between the tiedown brace and the barge deck, and that the tiedowns are arranged as inboard/outboard pairs. In this manner, tension that would have been developed in the tiedowns on one side of a support is added to the compression in the tiedowns on the opposite side.

Vertical support was provided by support cans attached between the cargo and the barge deck. These supports were modeled as beams and their loads are shown in the accompanying MOSES output in the reports titled "BEAM LOADS" and "BEAM ENVELOPE." The first of these gives the still water loads and the second gives the minimum and maximum overall load cases. Since these are beam loads, the vertical support load is the axial load in the beam and follows the standard convention where tension is positive. Therefore, any positive axial loads indicate uplift in the support.

As you can see, the installation macros made special load cases to keep track of the load signs. In this manner, we can ensure that any wave loads would increase the axial, shear, and bending moment of a member, not decrease it. We did not take this precaution in the native commands files. When you compare the results of the structural code checks, you will see that the results are different. You will also notice that the installation macros have taken the extra effort to take the tiedowns out of the section where the still water case is reported. In our native command file, the tiedowns are reported as part of the still water case as having 0 loads.

When you look at the graphical representation of the structural results, you will also see different colors. In the installation tools results the vertical members fail, this is in comparison to the native command where the vertical members pass.

We tried to make the installation tools easy to use. The purpose of this exercise is to show a comparison. A secondary purpose is to show the workings of the installation tools and show that the installation tools provide a rigorous method. You are welcome to use whichever method you prefer for your projects.

Exercise A

- Start with the deck from test files sac2 (see /ultra/hdesk/runs/tests/convert directory).
- Convert this file and transport the resulting deck on the Tidmar 251 (tdmar251.dat file).
- Take out the tiedowns that are translated with the model (elements with class T/D)
- Put the trailing nodes *J3304 and *J3303 at 100 ft from the bow.
- Put the bottom of the legs 5 ft above the barge deck.
- Use tubulars of OD 48 in ID 1.375 in for the cans.

- Use tubulars of OD 36 in ID 1.375 in for the tiedowns.
- Use the nodes *j3107, *j3108, *j3104, and *j3103 for tiedown connections at the deck.
- The tiedowns should go to the barge deck and shell intersection.
- The tiedowns should span 11 ft longitudinally between the deck point and the barge touch down point.
- Use the same env.dat for the fatigue data.
- The barge is to have a draft of 8 ft with a trim 0.57.
- Use the same winds and sea spectra.

The suggested cif and data files for this exercise are found in directory

/ultra/hdesk/runs/samples/install

files tow_exer.cif, tow_exer.dat. The translated deck file is in the directory

/ultra/hdesk/runs/samples/data

file dk_exer.dat.

6.2 Using the Orient Option

Topics

- Orienting the jacket to have one leg parallel to the barge sideshell.
- Calculating the hydrodynamic database with 3D diffraction.

Project Specifications

- Use the jacket in file quad.dat
- Use meters and mtons for reporting
- Use the Julieb barge
- Support the jacket on nodes *J0101 *J0201 *J0301 *J0401 *J0105 *J0205 *J0305 *J0405
- For the cans use OD 920mm and thickness 30mm
- Place node *J0101 15 meters from the bow and 17.5 meters on the starboard side.
- Place leg with nodes *J0101 and *J0401 parallel to the barge side shell.
- The centerline of the leg is 1.5 meters above the barge deck
- For the tiedowns use OD 420mm and thickness 25mm
- Use 4 tiedowns evenly distributed around each can
- For structural analysis assume barge can take tension from tiedowns
- Place the barge at 3.2 m draft and 0.57 degree trim
- Use the following ballast arrangement

name	%full	name	%full
2p	100	2s	100
3ap	100	3bp	100
3bs	100	4bp	100
4bs	100	5bs	60
5bp	100	6bp	100
6bs	100	8ap	100
8as	100	8c	100

- Use 3D diffraction to calculate the hydrodynamics
- Wind Values
- For intact stability use 100 knots
- For vortex shedding use 80 knots

- For structural analysis use 90 knots
- the environments to be used for calculating accelerations and structural analysis

name	Hs	Tm
D	1.7	10.5
F	2.3	8.9
G	1.2	7.0

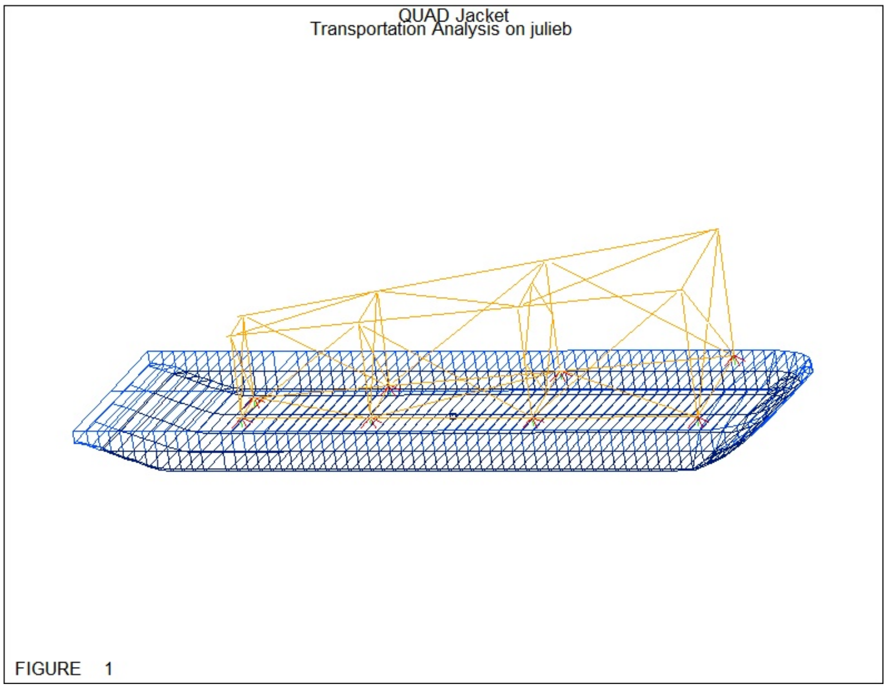


Figure 55: Isometric view of transportation

Discussion

Many of the project specifications we have seen before. There are three new specifications to be discussed.

- The ballast arrangment
- Use 3D diffraction to calculate the hydrodynamics
- Place leg with nodes *J0101 and *J0401 parallel to the barge side shell.

For this exercise, a translation from SACS is not needed. The MOSES model is provided. We will need to use a command file similar to ck_sac.cif to determine which nodes are at the top and bottom of each leg. We will also need the ck_sac.cif file to determine the order of the nodes. After clicking some we, find that the nodes at the ends of the legs are *J0101 and *J0401 we will call LEGA, *J0105 and *J0405 we will call LEGB. The instructions call for LEGA nodes to be parallel to the barge

side shell with a positive y coordinate. This means that LEGB will be on the port side. The nodes at the top of the jacket are *J0401 *J0403 *J0407 *J0405. The instructions tell us the distance from the bow to node *J0101 is 15 meters. This means the top of the jacket will be nearest the stern. To orient the jacket with one leg parallel to the centerline or sideshell, we use the *-orient* option. This is the same option we used for the deck with odd number of legs.

Figures 55 to 36 show the transportation configuration. The manual instruction for *-orient* are:

> For a structure which is not symmetrical about the barge centerline, the orientation scheme is different. Here, the *-orient* option is used. This option defines three nodes. The first node is where the distance for positioning will be measured and is normally at the bottom of the leg that is parallel with the deck edge, assuming the top of jacket faces aft. The second node is along the leg from the first node, and the third node is on the other side of the barge, usually along the horizontal level in line with the first node. Y is the distance from the centerline of the barge to the first node, positive towards starboard. Note that if one specifies starboard nodes and a negative Y then the jacket will be placed under the barge.

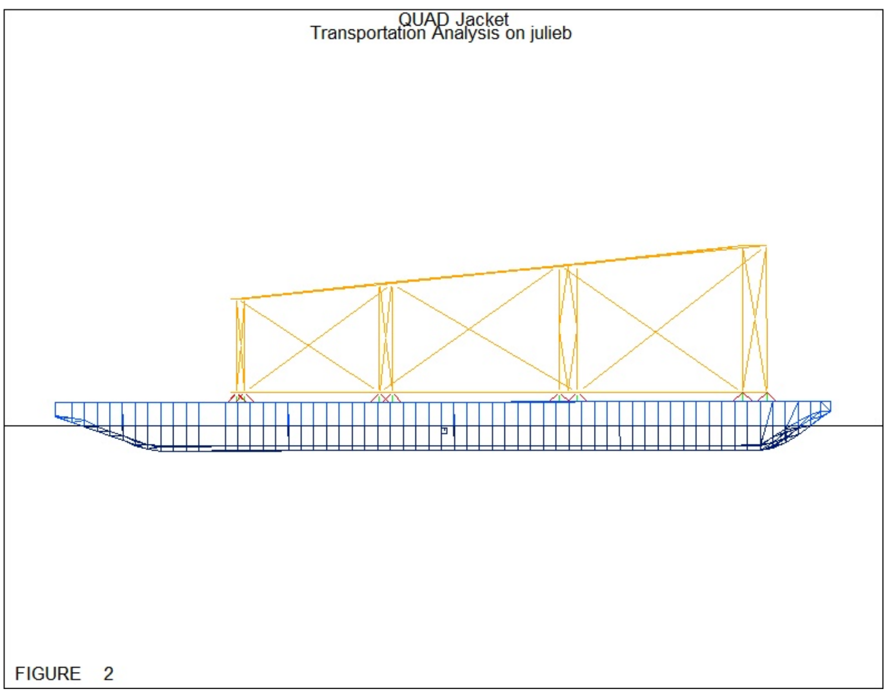


Figure 56: Side view of transportation

Notice we are using a negative sign and a port side node. This means our jacket will be on top of the barge. The command to place the barge in install.dat should read

something similar to:

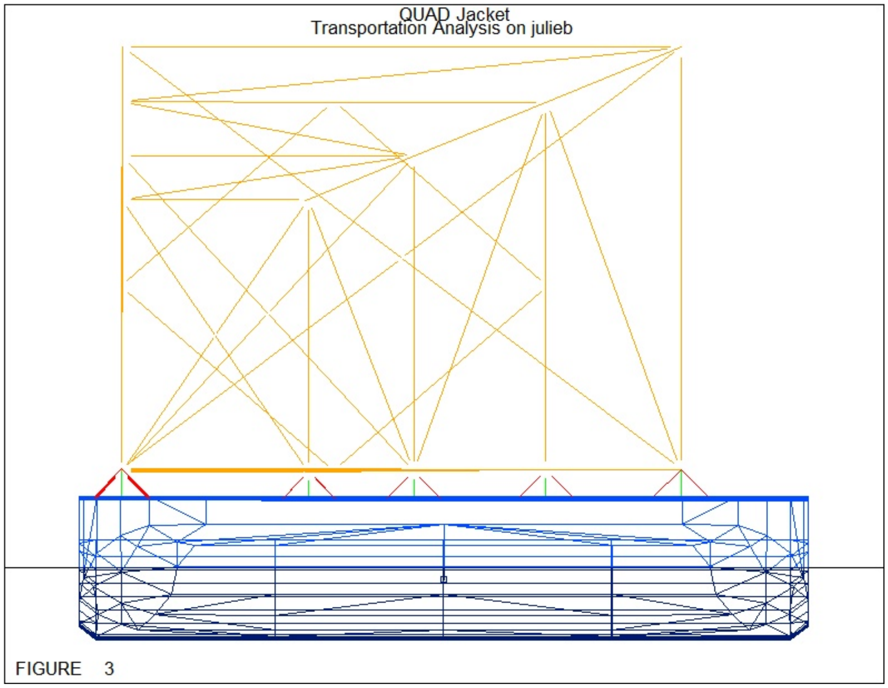


Figure 57: Bow view of transportation

```
$
$********************************************* barge data
$
use_ves julieb
$
$********************************************* structure data
$
&SET port_nod = *J0101 *J0201 *J0301 *J0401
&SET stbd_nod = *J0105 *J0205 *J0305 *J0405
$
$********************************************* define jacket
$
model_in tripod tripod.dat 15 17.5 1.5 \
   -port_nod      -stbd_nod      -orient *j0101 *j0401 *j0105
$
$********************************************* jacket/barge conn.
$
~tiedown tube 140 25
~vcan tube 320 30
i_connector v_can ~vcan %port_nod %stbd_nod
i_connector 4_tie ~tiedown %port_nod %stbd_nod
```

For the command file, all we need to do is add *&set d_type = 3ddif* before the inmodel

command. This will take care of the 3d-diffraction project specification.

Disscusion - Command File

The transportation part of the command file needs to use the *–ballast* option. We can make a selector for all the compartments that will be filled to 100%. It should read:
–ballast :full 100 5bp 95 6bp 95

Disscusion - Log and Output File

The big difference we are going to see for this analysis is in the run time. 3D-diffraction takes substantially longer time in comparison to strip theory. This will be evident, as you wait for the analysis, and in the log file MOSES reports every time it finishes the calculation for a wave period. Here is what the log file looks like when it calculates a 3d-diffraction database:

```
             Setting Pressure Name for JULIEB to D3.2
              ==========================
Time to Generate 494 Panels For JULIEB            : CP=   0.50
Time For 3D Diff.   494 Panels, Freq.   1         : CP=   2.71
Time For 3D Diff.   494 Panels, Freq.   2         : CP=   2.70
Time For 3D Diff.   494 Panels, Freq.   3         : CP=   2.68
Time For 3D Diff.   494 Panels, Freq.   4         : CP=   2.70
Time For 3D Diff.   494 Panels, Freq.   5         : CP=   2.71
Time For 3D Diff.   494 Panels, Freq.   6         : CP=   2.85
Time For 3D Diff.   494 Panels, Freq.   7         : CP=   3.05
Time For 3D Diff.   494 Panels, Freq.   8         : CP=   2.75
Time For 3D Diff.   494 Panels, Freq.   9         : CP=   2.74
Time For 3D Diff.   494 Panels, Freq.   10        : CP=   2.73
Time For 3D Diff.   494 Panels, Freq.   11        : CP=   2.74
Time For 3D Diff.   494 Panels, Freq.   12        : CP=   2.78
Time For 3D Diff.   494 Panels, Freq.   13        : CP=   2.80
Time For 3D Diff.   494 Panels, Freq.   14        : CP=   2.87
Time For 3D Diff.   494 Panels, Freq.   15        : CP=   3.09
Time For 3D Diff.   494 Panels, Freq.   16        : CP=   3.21
Time For 3D Diff.   494 Panels, Freq.   17        : CP=   4.08
              Setting Drift Name for JULIEB to D3.2
              ==========================
Time to Sum Pressures For 494 Panels on JULIEB    : CP=   0.19
Time To Set Up Convolution                        : CP=   0.03
```

When we review the pictures and the tiedown ends we see that some of the tiedowns did not land on the barge. Mathematically all the rigid connections are made and the analysis is completed. We do need to make these visual checks on the pictures

or the checks of the ends reporting to make sure our situation is real.

Other than that, the presentation of the log and output file is very similar to those we have seen before.

Exercise A

If you look closely at the tiedowns located at the stern starboard side you will see that they do not land on the barge. This needs to be corrected. We can use the *i_connector xy_delta* form of the automated installation command to fix this quick.

Change the bottom of data file, where the tiedowns are defined, to read.

i_connector v_can ~vcan %port_nod %stbd_nod
i_connector 4_tie ~tiedown %port_nod *j0201 *j0301 *j0401
i_connector xy_delta ~tiedown -1.5 1.5 *j0101
i_connector xy_delta ~tiedown 1.5 -1.5 *j0101
i_connector xy_delta ~tiedown -1.5 1 *j0101
i_connector xy_delta ~tiedown 1.5 1 *j0101

6.3 Launch

Introduction

This exercise presents a launch analysis in two methods. The analysis is done with native commands with a simple jacket model. Then a more real jacket model is used with the installation tools. The analysis done with the installation tools is considered more complete. The objective in presenting both methods is to show some of the steps that the tools are using, and to show that if you wanted to go "the long way," you could. Many of the native commands needed for the model summary reports were presented with the native transportation exercise. The discussion on those commands will not be repeated here.

For the native launch commands the files are under the tests/install directory. They are man_laun.cif, man_laun.dat, and ji_barge.dat. For the automated files we will use the same files used in the automated transportation exercise.

Discussion: Native Files

A launch analysis is a two body problem. The jacket body slides off the barge body. Much care needs to be taken so that the pre–launch condition is defined as the project desires. Most of the discussion here for the "long way" concentrates on setting up the pre–launch condition.

For this exercise, a translation from SACS is not needed. The MOSES model is provided. There is a restraint at the end of the file that is needed.

```
~SPRING FIX    REST ~SPRING *J0001
```

This restraint is needed so that the structural solution will solve.

We will need to use a command file similar to ck_sac.cif to determine which nodes are at the top and bottom of each leg. We will also need the ck_sac.cif file to determine the order of the nodes. This is a very basic jacket. The leg with nodes *J0006, *J0004 we will call LEGA, the leg with nodes *J0003 and *J0001 we will call LEGB. Further lets disignate nodes *J0006 and *J0003 as the top of the jacket. This end of the jacket is the leading edge. Meaning it leads into the water. Nodes *J0004 and *J0001 are the trailing edge. Meaning they go into the water last. Now back to the command file.

The first part of the command file should be familiar to us. The new part is in the model editing portion. You are familiar with the command *medit* from previous exercises. This is the first time we will be defining a launchway assembly. The

following is the link to the manual page for the command *assembly LLEG*.

bentley.ultramarine.com/hdesk/ref_man/conn_lway.htm

For this command it would be beneficial if you took the time to read the entire page before starting to look at the specifics of the format. There is a figure at the top of the page that is also shown here, Figure 58.

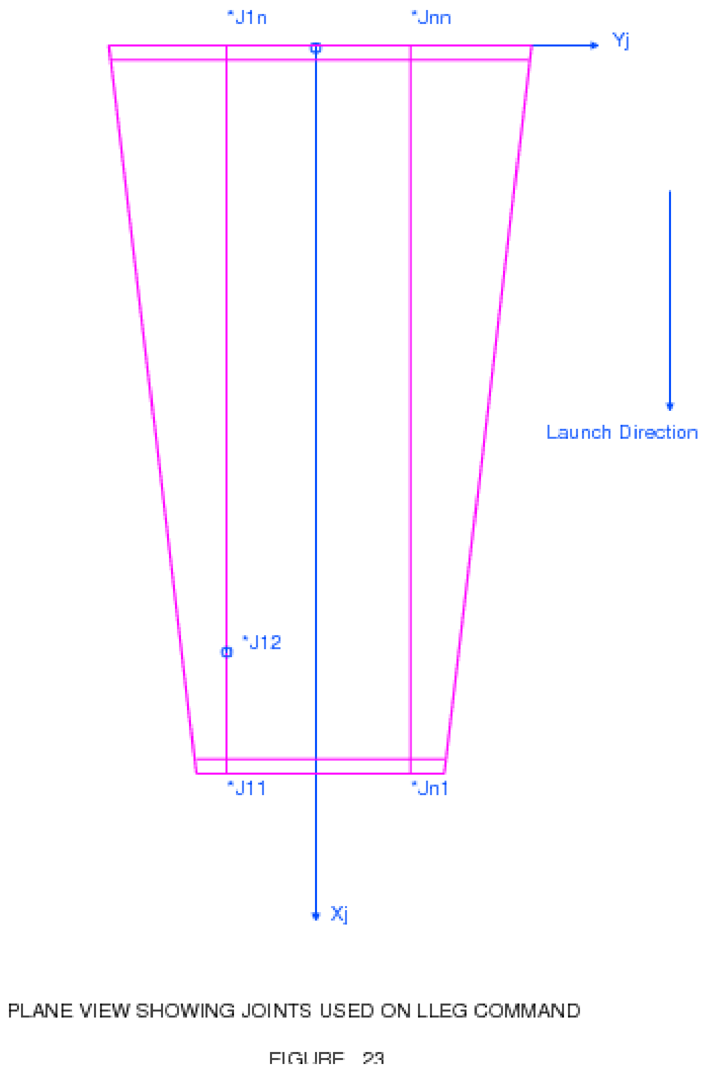


Figure 58: Joints used in the Assembly LLEG command

There is a second figure showing the tiltbeam geometry towards the bottom of the page, and is shown here as Figure 59, Tiltbeam Geometry.

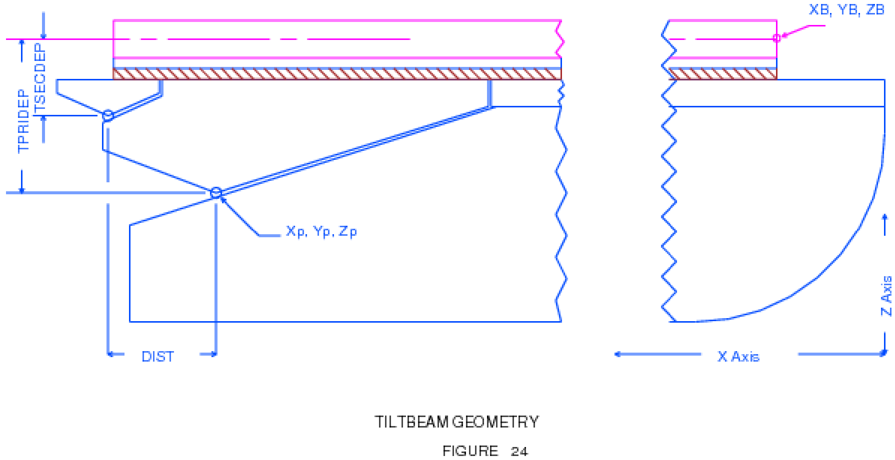


Figure 59: Tiltbeam Geometry

The following table presents the legs in the various forms they will be labeled and associated. Keep this table in mind when we are discussing the *assembly lway* command.

Label	Leg A	Leg B
Barge side	starboard	port
Leading node	J1003	J1006
Trailing node	J0001	J0004
Assembly list	first	last

Table 5: Leg Name shows various labels

We are going to discuss the needed data as it appears in the *assembly lleg* command. Near the top of the manual page there is a top view of the launch barge assembly. From this sketch you can see the map between the order of the nodes on the command and the location on the barge assembly. Also, as part of the description we find that:

> Here, ∗J(1) . . . ∗J(n) are the node names of the nodes along the launch cradle of the jacket, in order, where ∗J(1) is the first node which will enter the water and ∗J(n) is the last node which will enter the water.

This refers back to the leading edge and trailing edge. The rest of the paragraphs helps fill out more of the command.

> The launch cradle is considered part of the jacket that rests on the barge skidway. BODY_NAME(1) is the body name assigned to the barge where the tiltpins are attached, and XB, YB, and ZB are the coordinates, in the BODY_NAME(1) body system, of the begining of the skidway on the BODY_NAME(1). Here, the skidway should be considered to be at the height of the jacket launch leg centerline above the barge origin. Also, :B1(i) are the selectors for the nodes the BODY_NAME(1) which will be used for connecting the jacket to the barge body.

We know our barge body name is "barge" but now we need to go to the figure towards the bottom of the page to get a better idea of what values to put in for XB, YB, and ZB. From this figure we see that XB, YB, and ZB are locating the trailing edge nodes on the barge. Please note that they are locating the centerline of the launch leg. The values used in the command file differ only by a sign on the YB value. The starboard side leg has a positive and the port side leg has a negative.

The last part of the command requests a selector :B1(i), however we have put ∗B@T and ∗B@. We could have used a selector to pic the top nodes on the barge, or we could have made a selector to pick the nearest barge node. This is an advanced exercise, we see that for this command it is acceptable to shortcut a selector for a search with a wild character.

The launch leg will search for the nearest barge node to make the connection. The launch leg connector has to make this determination for each step during the launch. Each time the jacket moves the connection points have to re–evaluated. Depending on the location of the jacket the connectors are turned on and off. For the starboard leg it will look for the nodes ending in "T". For the port leg it will include all the barge nodes. It will however pick nodes ending in "T", because those nodes are at the top of the barge, and nearest the leg.

For the options *–TPIN* we need to concentrate on the lower figure. This option is locating the tilt pin. Please pay close attention that "TPRIDEP" is to the centerline of the launch leg. XP, YP, and ZP are in the barge, BODY_NAME, coordinate system. The last two options *–FRICT* and *–BEAM* are for the coefficient of friction and the section properties of the tiltbeam.

We still need to establish the coordinate system for launch. Here we need to read the 2nd paragraph under the Tiltbeam Geometry figure.

> The order of the input of the *ASSEMBLY LLEG* command is important as it is used to establish the launch coordinate system of the jacket. The axes of this coordinate system are set as follows: The X axis is parallel to a line connecting ∗J(1) and ∗J(n), and is directed towards ∗J(1). The jacket is launched in the positive X direction. The origin of the system is midway between the trailing joints given on the first and last *ASSEMBLY LLEG* command, the Y axis is along the line connecting the ∗J(n) on the last *ASSEMBLY LLEG* input with the ∗J(n) on the first one input. The Z axis is determined from the right hand rule.

Remember the table at the begining of this discussion? It is the starboard leg that should be defined before the port leg if the launch direction is towards the stern.

That concludes the model editing needed for launch.

In the automated method a section of the output is dedicated to reporting the model. Here we are just going to report the summary of the restraints, and a short category

summary of the jacket.

&summ
 restraint
 cat –part jacket
end

It is a good idea to run the analysis and have the log and output files available for discussion.

The restraint report contains some unexpected entries. We defined the launchway assembly with the command *assembly lleg*, the first row of the report shows an element with name &LR1-001 and a class of &LRUNNER connected to to ∗J0006. A launchway connector is a really a complicated connector. The ends of the connector on the barge have to change and eventually remain turned off when the jacket enters the water. Having the user input all these connector specifications would be very time consuming and prone to human error. This is the default naming and numbering scheme done within MOSES.

The last row shows the spring restraint that was part of the data file.

The Category Summary for Part Jacket you should be familiar with. A similar report was presented as part of the transportation exercise.

The next set of commands you may be mostly familiar with.

&INSTATE barge –condition 0 0 3
&weight –compute barge 10 32 0.29*330 0.29*330
&status config
&status g_lway
&picture side

The only new command is *&status g_lway*. This command generates the table Launch Way Geometry that is found in the log file. When you review this file you will see many of the inputs that were used in the

This is a good place to do Exercise A.

Performing the Launch
The next command actually performs the launch.

LAUNCH –MAXTIM 100 –MAXOSC 5 –TSTEP .5 .25 .5 -WINCH .5

This command performs the launch simulation. The log file contains a small summary of the analysis, it is titled "Launch Events Summary". The options used are presented in the order that they are used in the simulation. The option *–WINCH* tells moses to assume the winches can move the jacket .5 m/sec until the jacket starts sliding. The log file shows that sliding occured at 0.5 seconds. Tip occus at 30 seconds and

separation occurs at 37.25 seconds. The oscillations are at 54.25, 60.25, 66.25 and 72.75 seconds. The simulations ends at 72.5 seconds because this is the 4th oscillation and the option *–MAXOSC* told MOSES to stop before 5. The option *–MAXTIME* which designates the final simulation times was not used. Had the oscillations taken longer to occur the maxtime may have been reached.

In the log file, the table "Launch Events Summary" has columns "Time" and "Change in Time", which seem to report at various intervals but not what is designated in the *–TSTEP* option. To see the effects of th *–TSTEP* option we need to go to the output file. The report "Location of the Origin of Selected Bodies" was produced by the commands:

```
PRCPOST
 LAUP_STD
END
```

If we focus on the "Event" column we see the time step increases in increments of 0.5 for the first 30 seconds, or until tip. The time step increases by 0.25 seconds until 37.25, or until separation. Then it goes back to increaseing by 0.5 seconds until the simulation ends. This is what was specifiec by the option *–TSTEP*. The first value indicates the time step before tipping, the 2nd value indicates the time step between tipping and separation, and the 3rd value indicates the time step after separation. Each one of these time steps can be created into a load case and used in a structural analysis.

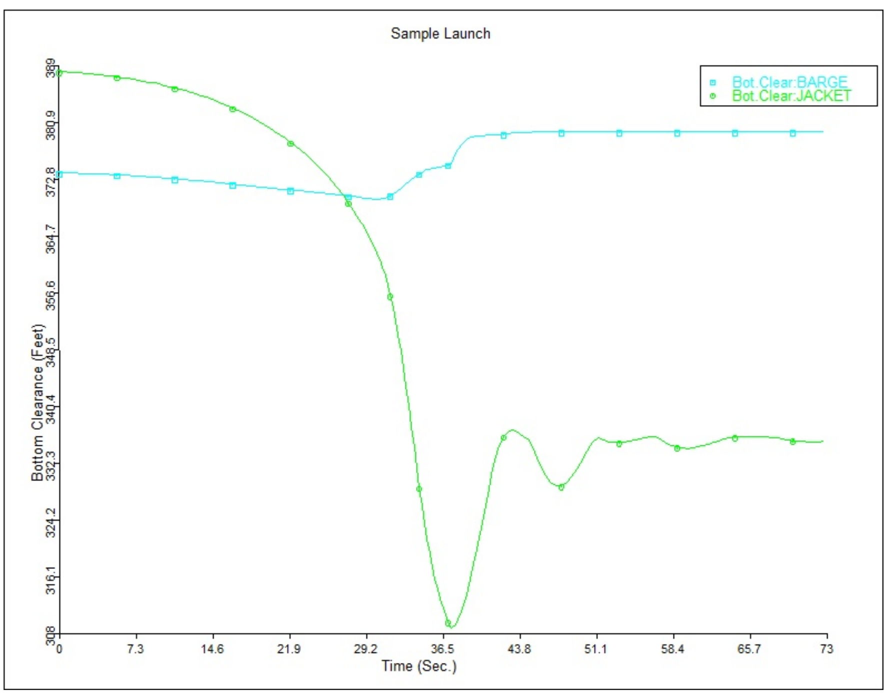


Figure 60: Barge and Jacket Clearance

Usually the time step between tipping and separation is lower than the others. This is becuase during tipping a higher percentage of the jacket weight is being supported by the rocker arm. By having more time events during this short time more load cases can be defined.

The command *LAUP_STD* produces a standard set of output for the launch analysis. This is actually a quick way of producing LAUNch Process STanDard output. The reports it creates is "Location of the Origin of Selected Bodies", "Velocity of the Origin of Selected Bodies", and "Skidway Reacations" reports. It also creates the plots in the answers directory. These are the plots that we have noticed many MOSES users find helpful in evaluating the launch process. Certainly, knowing the bottom clearance is helpful, as shown in Figure 60: Barge and Jacket Clearance. Knowing the jacket slides off the stern instead of tipping off a side is helpful, as is shown in Figure 61, Vertical Force at Pin and Trailing Edge is helpful.

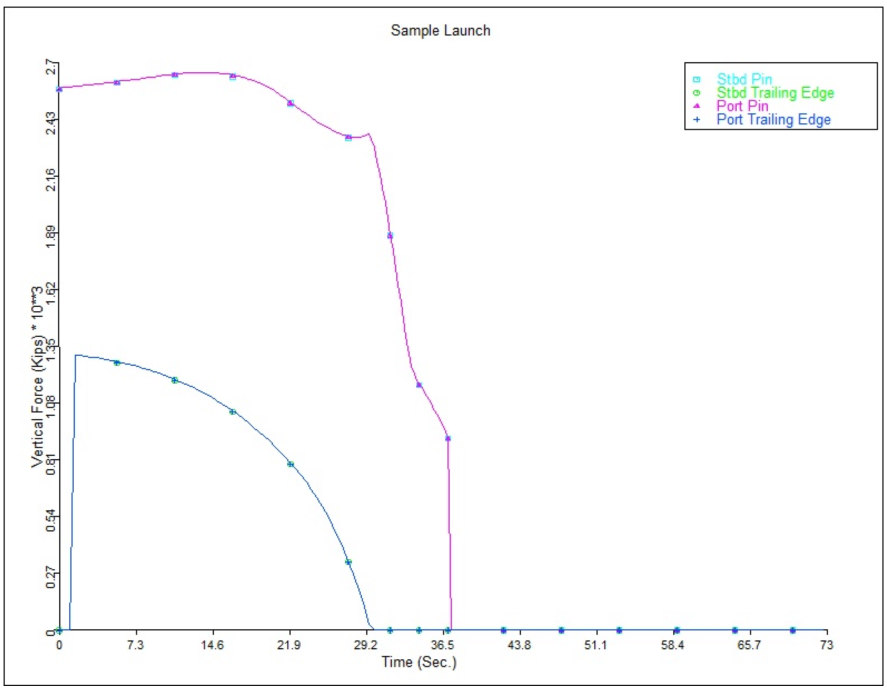


Figure 61: Vertical Force at Pin and Trailing Edge

In the next long section we solve the structural solution three different ways. The three ways are: Approximate Solution, Better Solution Before Tip, and Combining Both. You might recall the general format for the Structural menu from the Longitudinal Strength Part 2 exercise. Because this is a two body body problem and

becuase we are using launchway connectors we have a few more commands.

In all three solutions we enter the structural menu with *structral* and exit the strucutral menu with *end*. For all three solutions we create a short output in the structural post processing menu. We enter this menu with the command *STRPOST* and exit it with the command *end*. The command we are going to be focusing on is *LCASE*. It can be found at the following link.

bentley.ultramarine.com/hdesk/ref_man/str_lcas.htm

For the structural solution we are going to focus on time steps: 0, 25, 30, 35, 37 and 40. Two before tip, three between tip and separation, and one after separation.

For the approximate solution we have:

```
LCASE –lleforce 1. 2. 50 -process 0. 25 30 35 27 40
S_PART JACKET
S_REST ˜SPRING
SSOLVE
```

The manual tells us that:

> Before tipping, a single distribution will be generated. After tipping, the distribution will be composed of two trapezoidal distributions, each TBLEN (feet or meters) long, which are symmetric about the tiltpin. The relative intensities at the pin and at the ends of each distribution are goverend by the two parameters QBEG and QMID.

The load distribution between tipping and separation is respresented by the rocker arm load in Figure 62: Rocker Arm Lad Distribution.

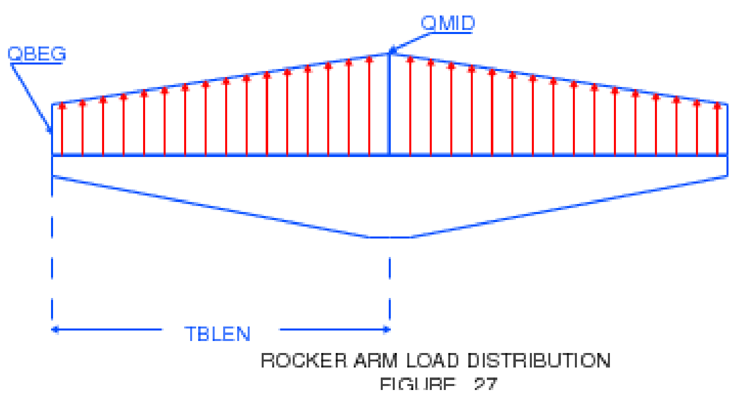


Figure 62: Rocker Arm Load Distribution

For the better before tip solution we have:

```
LCASE -process 0. 25 30 35 27 40
S_BODY JACKET
S_REST ˜&LWAY
SSOLVE -NONLINEAR
```

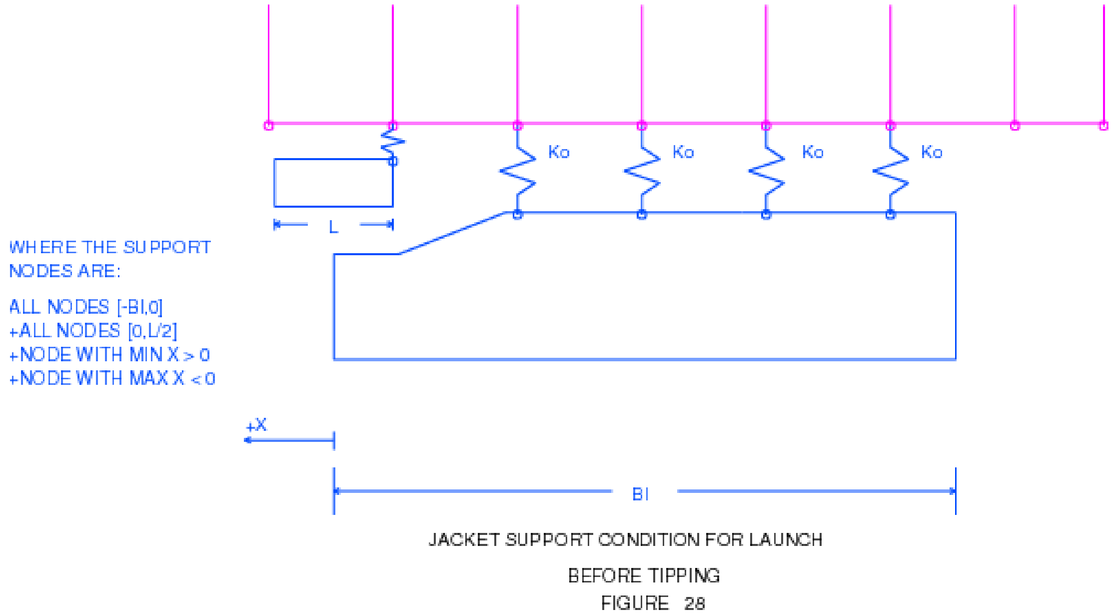


Figure 63: Jacket Support Condition Before Tip

The manual tells us that:

> If either *SSOLVE –NONLINEAR* or *BODSOLVE* is used for solving a launch, MOSES will create connections modeling the launchway at each load case. The type of restraint supplied in the area of the tiltbeam is dependent on the use of the *-BEAM* option of the *LLEG* command. This option defines the bending stiffness and length of the beam. Before tipping, all jacket nodes between the two ends of the tiltbeam will be restrained. In cases where this criteria does not yield at least two nodes, the node furthest forward yet aft of the pin and the node furthest aft yet forward of the pin will be added to the others. The selected nodes are then connected with compression only springs to the closest barge nodes. For the nodes in contact with the barge proper, a nominal "stiff" spring is used. For those in contact with the tiltbeam, the stiff spring is put in series with the bending stiffness of the tiltbeam at the proper location.

The load distribution represented by the use of *LCASE* in this manner is shown in Figure 63: Jacket Support Condition Before Tip and Figure 43: Jacket Support

Condition After Tipping.

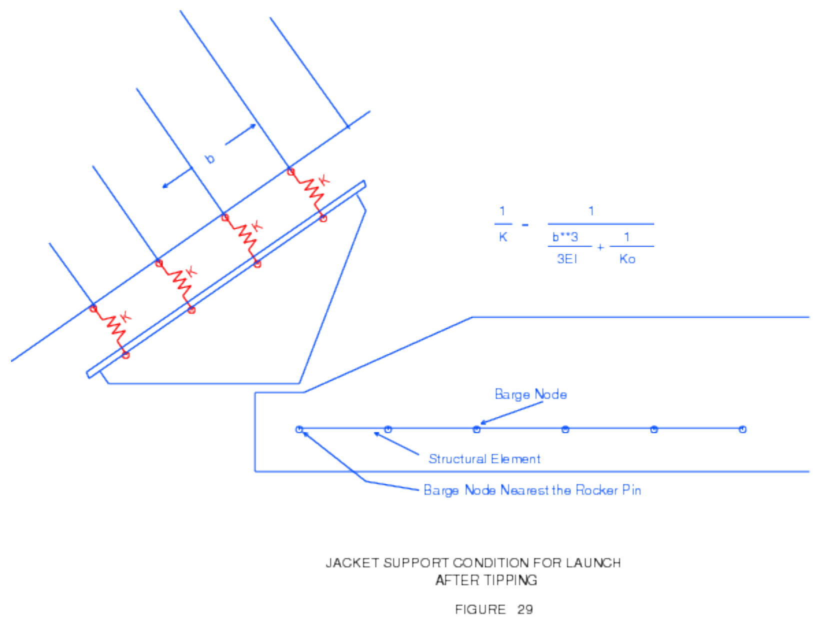


Figure 64: Jacket Support Condition After Tipping

As you can see we really need a compromise here. We need to combine both methods. This is what is done in the "Combining Both" section.

```
LCASE -process 0. 25
LCASE -lleforce 1. 2. 50 -process 30 35 27 40
S_BODY JACKET
S_REST ~&LWAY
SSOLVE -NONLINEAR
```

Notice that the output for each structural analysis is similar. For the approximate solution only one restraint report is generated. Remember for the approximate solution only the ~SPRING restraint is used. The table summarizes the value of this restraint for the five load cases generated. For the other two methods each &LR element can be a restraint at each time step. A separate table is needed to represent the restraints at each time step.

This concludes the discussion of the native files for launch. The following exercise concentrates on understanding the output.

Exercise A

It is a good idea to take the values from the *&status g_lway* table and place them in

their corresponding location in Figure 64.

Exercise B

Fill in the table with the values presented in the structural WS Beam Check Standard reports. “BB tip” stands for Better Before Tip method.

Element	Method	Case	Unity Ratio
HOR|0068	Approximate BB Tip Combined		
VER|0074	Approximate BB Tip Combined		

Table 6: Comparison of Structural Results

Discussion: Automated Files

For the automated files (install.cif, install.dat, big_jack.dat, and env.dat), we are going to use the installation files that are in the sample directory. They can be found at the following link.

bentley.ultramarine.com/hdesk/runs/samples/install/list.htm

The use of these files is also found at the following link.

bentley.ultramarine.com/hdesk/runs/samples/install/launch.htm

bentley.ultramarine.com/hdesk/runs/samples/install/transp.htm

These files are very similar to the tow_auto files in the transportation exercise. When you review the general format of the install.dat file you will see that it is basically the same as tow_auto.dat. The main difference is that now we are using a real jacket and therefore more nodes are needed to describe the port_nod and stbd_nod variables. The other difference is that we are now using *i_connector v_lway* instead of *i_connector v_can*. The command *i_connector v_lway* defines the launchway assembly based on the nodes used with the option *–port_node* and *–stbd_node*. This one command certainly saves us much typing.

The versatility of these files is that you can use them for launch, transportation, loadout, upand and lift. You can run all 5 analysis successively or run a subset. For our discussion we are going to only run transportation and launch. In the cif file we want to leave the value of the "launch" and "transportation" variables to ".true.". This is lines 8 through 12.

&set launch = .true.
&set transportation = .true.
&set loadout = .false.
&set upend = .false.
&set lift = .false.

This will run the transportation and the launch analysis in one MOSES session. The automated files will take care to define one body consisting of the barge and jacket part for the transportation analysis. Separately, it will define a body jacket and a body barge for the launch analysis. The final strucutral analysis results will report the worst offending loadcase. It will not discriminate between a transporation loadcase or a launch loadcase, it simply reports the worst offender. This helps in evaluating what part of the installation process is causeing high unity ratio.

Here are a few views of the launch pre–condition.

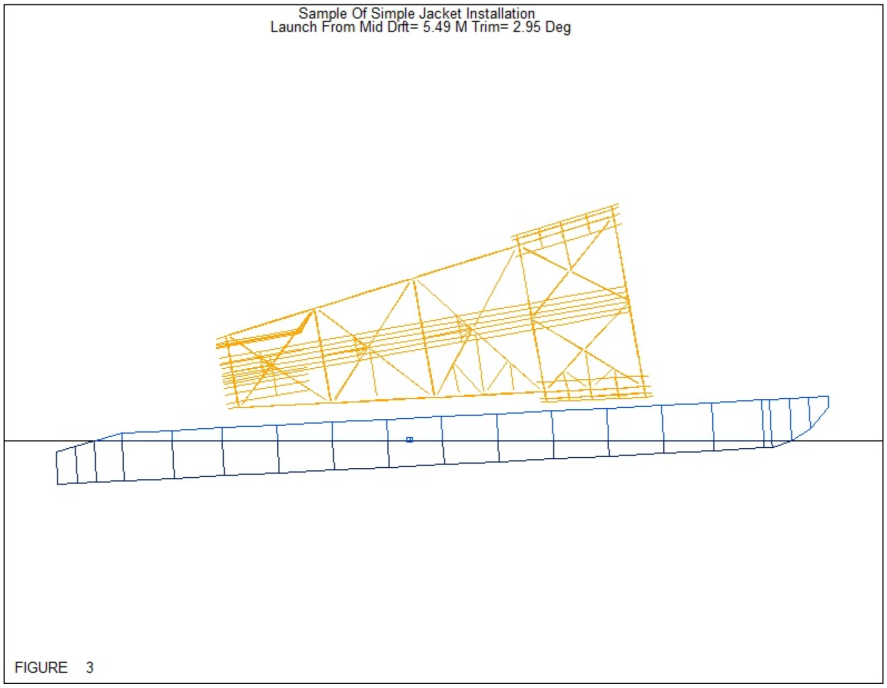


Figure 65: Side view of Pre-Launch Condition

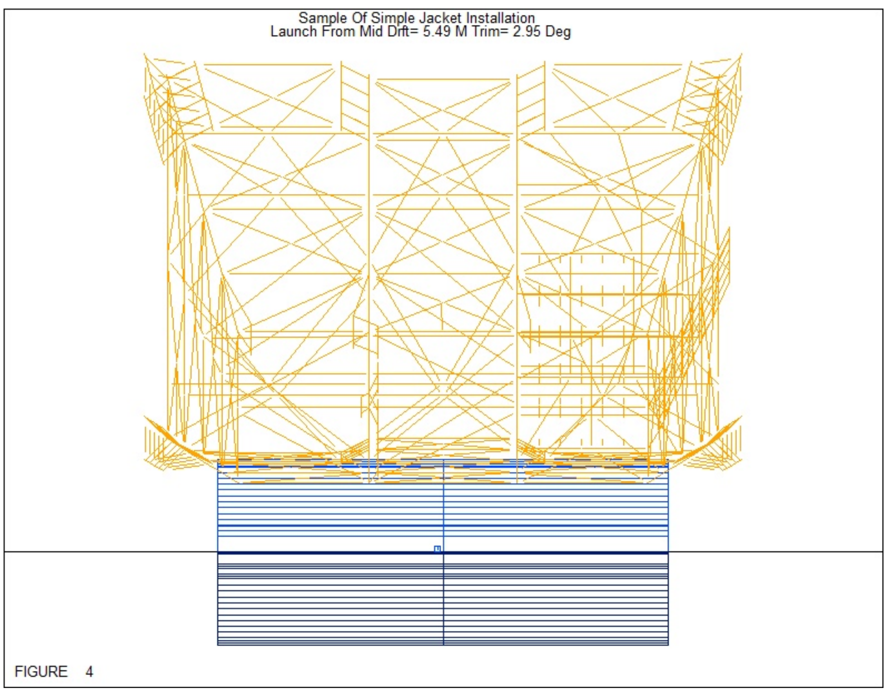


Figure 66: Bow view of Pre-Launch Condition

For the launch portion of the cif file all we need is the *INST_LAUNCH* command and some of the options. For the launch we are doing here we will be setting some of the same values we set in the "long method", but with values appropriate for this barge

and jacket.

```
&if %launch &then
 inst_launch –friction .06 –draft 5.49 –trim 2.95 \
  –nonlinear –winch .5
&endif
```

Just like in the "long method" the friction and the winch speed are defined. For the "long method" the condition of the barge was set with *&instate -condition*, here we are using the *–draft* and *–trim* options. Please be careful and read the instructions. For this option the draft at midships is being defined. The last option, *–nonlinear*, tells MOSES how to model the launch way connector during the strctural analysis.

This is all that is needed in the automated method. Here are a fews of the standard plots that are created.

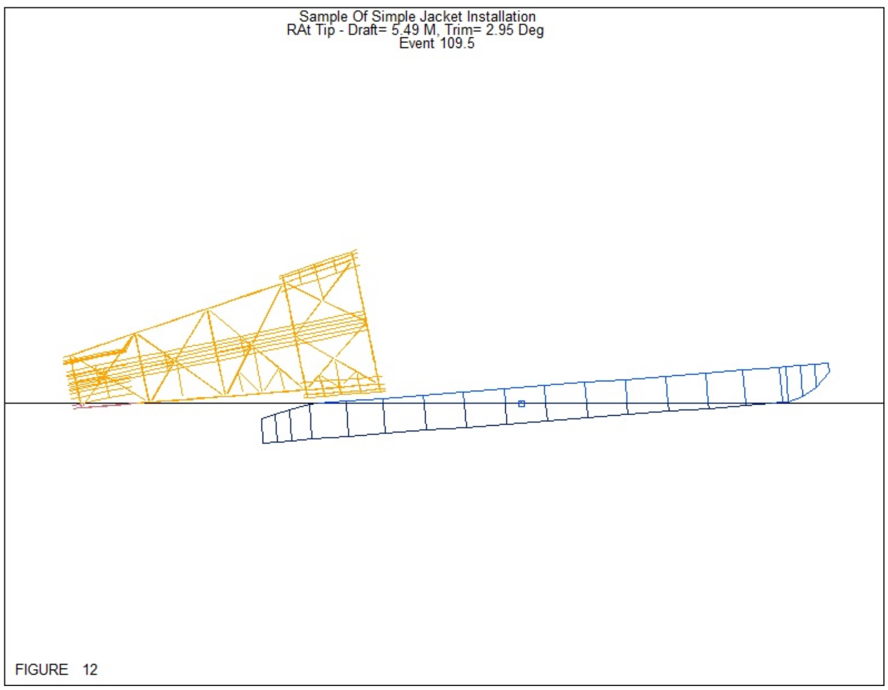


Figure 67: Side view of Tip Event

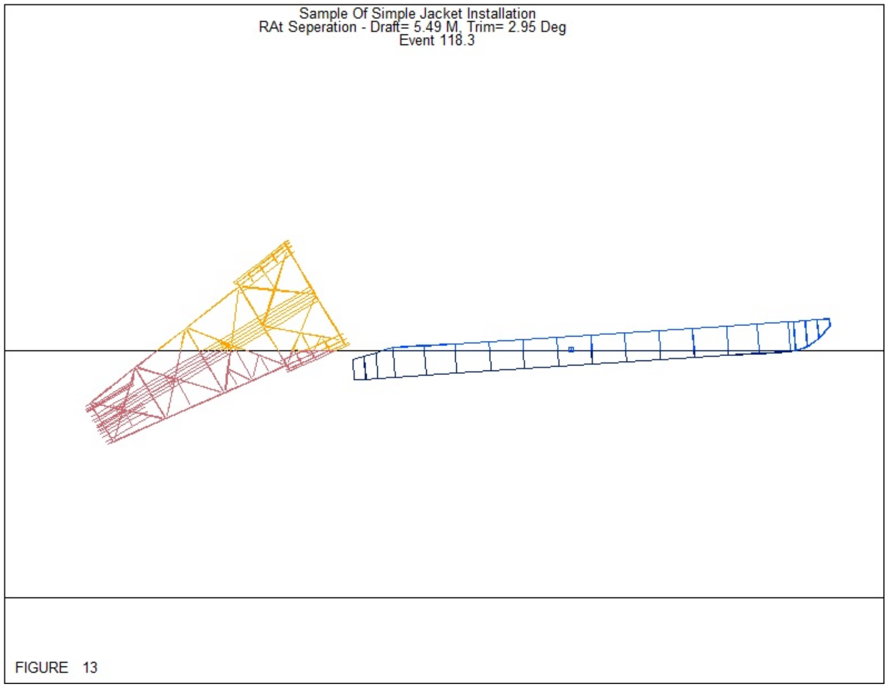


Figure 68: Side view of Separation Event

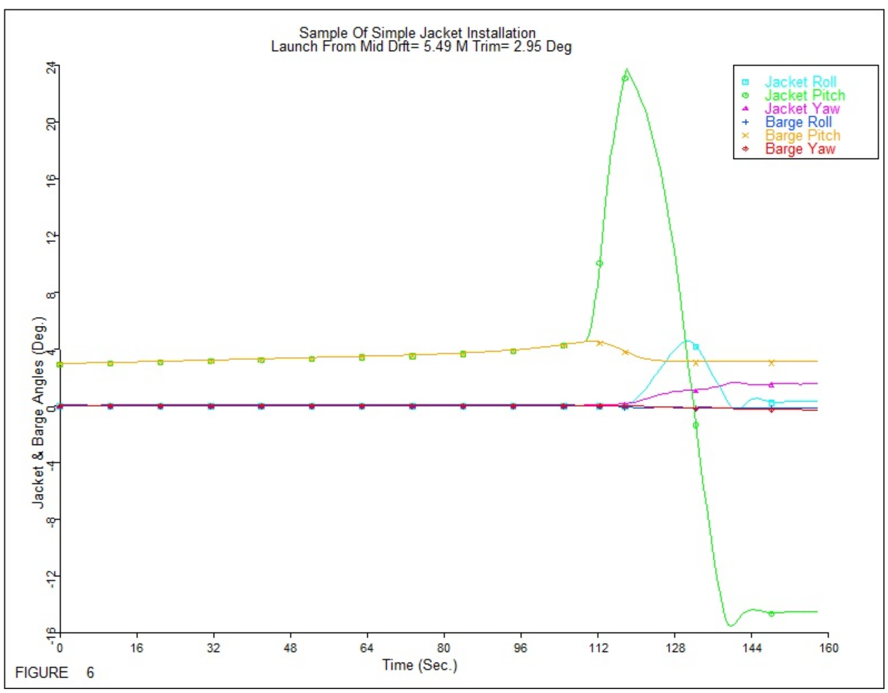


Figure 69: Trajectory plot of two bodies

Notice on the Rocker Arm Reactions plot that the port side leg leave the barge at a different time than the starboard side leg. When we review the output Page 53, Skidway Reactions report, we see that the difference is 1 second (108.5 – starboard side, 109.5 – port side). We would have to present this to the project to determine if

this second delay is acceptable.

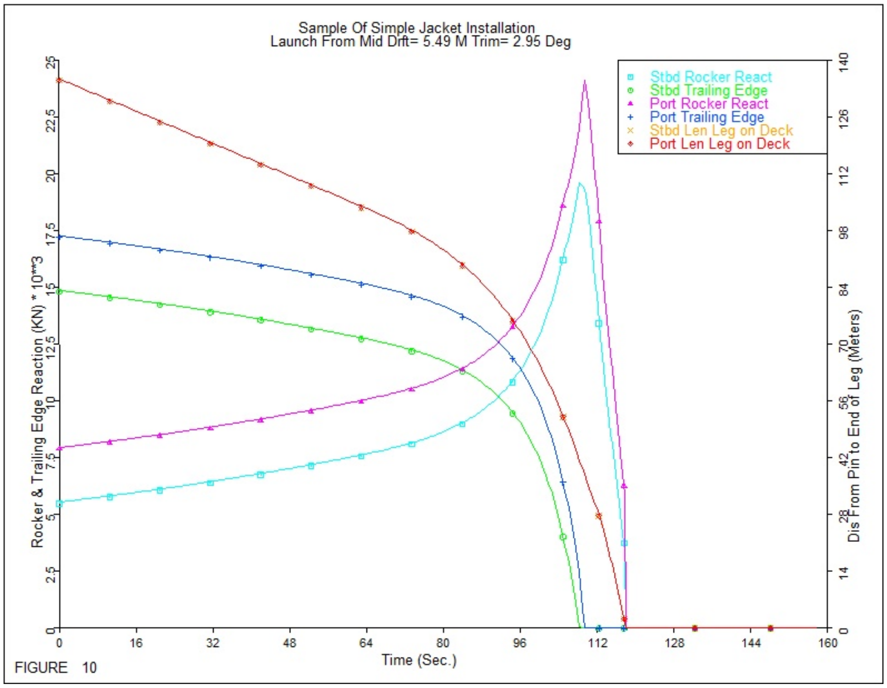


Figure 70: Rocker Arm Reactions and Distance from Pin

7 Answers

7.1 Getting Started Exercise

1	KML for 12 ft draft = 1117.07 ft, and
	KMT for 12 ft draft = 75.44 ft
2	Righting arm = 17.84 ft for 28 deg roll
3	At longitudinal location = 0, 400 and 200 ft
4	At longitudinal location = 0, 400 and 200 ft
NOTE	Need to review table headers to understand what is being reported

7.2 Basic Stability Exercise

BSTAB		
	1	379.30 ft
	2	78.13 ft
	3	Command 'RARM 2.5 10' was used
	4	The dat file is not echoed in the out00001.txt file
WCOMP		
	1	938.1 kips
	2	5P and 5S
	3	30.10
Exercise		
	1	Trans = 170 ft^4 Long = 227 ft^4
	2	Area ratio = 5.31

7.3 Free Surface Correction Exercise

1. 0.31 m^2

7.4 Compartment Ballasting Exercise

1. The answers do not change. The area of the valve is not important in a static analysis.

7.5 Stability Check and KG Allow

1. Passes

2. Passes

3. For Draft 6, Allowable KG is 14.13, Area Ratio, Yaw = 0, damage none! Controls

7.6 Review Working with Compartments

1. Passes

2. Passes
3. 2644.61 Kips
4. 4.17 feet
5. 5s, 5p
6. 5s, 5p and 5c
7. 3c

7.7 Dynamic Flooding

Exercise A

1. 9 - RY:TBRG
2. x-axis, Event
3. left axis, Z:TBRG
4. right axis, RY:TBRG
5. plot 1 4 -rax 3 -no

Exercise B

1. Simulation Terminated Due to Capsizing has changed to Simulation Terminated at Specified Time
2. out0001 column Intern. Fl. Head and Vlv Diff Head is blank for the one with total time 1200 sec column Intern. Fl. Head and Vlv Diff Head is full for the one with total time 200 sec

Exercise C The command line before *tdom* should now read

1. &compartment -correct two -percent two 0 -dynam two

7.8 Basic Frequency Domain

Exercise B

1. Yes, the righting arm and wind arm have changed. Before the changes the righting arm crossed zero the second time after 67 degrees. After the change the second zero crossing is around 54 deg.
2. Yes, the draft and pitch changed. The draft is now deeper and the pitch is less.

7.9 Modeling Cargo Exercise

Exercise A

1. 0.901
2. 800.57 kips

Exercise B

1. Fails
2. minimum GM
3. 2037 KN
4. 68.99 KN

Exercise C
The answers for these questions may differ from those presented here. The answers are dependent on the ballast arrangement. These answers were determined with the ballast arrangement:

Name	% Full
4P	96.13
4S	73.5
5C	53.4
5P	100
5S	64.82

Answers

1. 6.90 KN
2. 4.11 KN
3. Yes
4. 16796KN
5. 0.072
6. 0.165
7. 0.299

7.10 Translating from SACS Exercise

Exercise A

1. 215. kN
2. x = 0 m, y = -13.79 m, z = -45.91 m

3. 48328 kN

Exercise B

1. x = 11.26 ft, y = -130.13 ft, z = -13.10 ft
2. 1553 kips

7.11 Longitudinal Strength Exercise

Exercise A
The command *&equi* moves all six degrees of freedom. The command *equi_h* move only z, rx, and ry.
Exercise B
The shear force and bending moment have small changes compared to the original results.
Exercise C
63.32 kips/ft
Exercise D
The shear force and bending moment from weight distribution in exercise C more closely resembles that in the original set up using *&weight -compute*.

7.12 Modeling a Fender

Copy the fender.cif command to another root name, for this exercise I will use f_exer.cif. Insert the line *&device -auxin fender.dat* before the *inmodel* command so that you can use the same data file fender.dat.
Change the *&instate -locate* command for the barge to the following
   -LOCATE barge 171 29.35 -4.3 0 0 90
Change the attachements for the tanker body to the following locations.
   *fent1 161 29.35 21.95
   *fent2 166 29.35 21.95
   *fent3 176 29.35 21.95
   *fent4 181 29.35 21.95
Change the attachment for the barge body to the following locations.
   *fenb1 0 10 4.3
   *fenb2 0 5 4.3
   *fenb3 0 -5 4.3
   *fenb4 00 -10 4.3
Change the attachment point order and change the euler angle for the connectors.
   connector f1 fend *fenb1 *fent1 -euler 0 0 180
   connector f2 fend *fenb2 *fent2 -euler 0 0 180
   connector f3 fend *fenb3 *fent3 -euler 0 0 180
   connector f4 fend *fenb4 *fent4 -euler 0 0 180

7.13 Tripod Jacket Transportation

Exercise B

Joint JW110 is where the leg changes outer diameter.